

BAN CƠ YẾU CHÍNH PHỦ

HỌC VIỆN KỸ THUẬT MẬT MÃ

¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯

**BÁO CÁO KẾT THÚC HỌC PHẦN LẬP TRÌNH C#**

Hệ thống quản lý cửa hàng cafe F&B

Ngành: Công nghệ phần mềm

Mã số:

*Sinh viên thực hiện*:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã sinh viên | Họ và tên | Lớp | Email |
| 1 | CT06N012 | Cấy Triều Đông | CT06CN1 | ct06n0112@actvn.edu.vn |
| 2 | CT07N0111 | Mạch Tiến Duy | CT07CN1 | ct07n0111@actvn.edu.vn |
| 3 | CT07N0162 | Lê Thanh Yên | CT07CN1 | ct07n0162@actvn.edu.vn |

*Giảng viên hướng dẫn:*

Nguyễn Văn Hoàng

TP. Hồ Chí Minh – 09/2025

BAN CƠ YẾU CHÍNH PHỦ

**HỌC VIỆN KỸ THUẬT MẬT MÃ**

**ĐÁNH GIÁ NHẬN XÉT CỦA GIẢNG VIÊN**

Họ và tên giảng viên: ***Th.S Nguyễn Văn Hoàng***

Tên đề tài: ***Hệ thống quản lý cửa hàng cafe F&B***

Nội dung đánh giá nhận xét:

……, ngày …… tháng … năm 20… Người nhận xét

(Ký và ghi rõ họ tên)

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin gửi lời cảm ơn sâu sắc đến Thầy Th.S Nguyễn Văn Hoàng đã tận tình hướng dẫn và hỗ trợ chúng em trong quá trình thực hiện đồ án môn Lập trình C#.

Trong quá trình xây dựng và hoàn thiện báo cáo cũng như xấy dựng ứng dụng theo đề tài đã chọn, nhờ sự chỉ dẫn tận tâm, những kinh nghiệm quý báu và những lời khuyên hữu ích của Thầy, chúng em đã hoàn thành đồ án một cách hiệu quả và có được nhiều kiến thức bổ ích.

Chúng em xin chân thành cảm ơn Thầy!

**GIỚI THIỆU**

Trong bối cảnh công nghệ ngày càng phát triển, ngành F&B (Food & Beverage) đối mặt với nhu cầu quản lý hiệu quả cửa hàng, từ tồn kho, nhân sự, bán hàng đến chăm sóc khách hàng. Việc áp dụng các ứng dụng quản lý thông minh không chỉ giúp tiết kiệm thời gian, tối ưu hóa nguồn lực mà còn nâng cao trải nghiệm khách hàng, tạo lợi thế cạnh tranh trên thị trường. Chính vì vậy, đề tài **“Lập trình ứng dụng quản lý cửa hàng cafe F&B”** mang thực tiễn, hướng tới xây dựng hệ thống phần mềm hỗ trợ các chủ cửa hàng quản lý hoạt động kinh doanh hiệu quả.

Ứng dụng được phát triển trên nền tảng winform C#, tích hợp các chức năng quản lý sản phẩm, tồn kho, nhân viên, doanh thu và báo cáo thống kê chi tiết. Việc triển khai đề tài không chỉ giúp nhóm sinh viên rèn luyện kỹ năng lập trình và thiết kế phần mềm, mà còn có cơ hội tiếp cận thực tế.

Bên cạnh đó, thành viên trong từng có trải nghiệm làm việc tại một cửa hàng cà phê có sử dụng hệ thống quản lý F&B: hệ thống giúp xử lý order nhanh, quản lý tồn kho chính xác và hỗ trợ doanh thu tức thì, nhưng cũng bộc lộ hạn chế như tính năng rời rạc và thiếu báo cáo phân tích. Từ trải nghiệm đó, chúng em càng nhận thấy rõ nhu cầu cấp thiết của một ứng dụng quản lý cửa hàng hiện đại, đầy đủ và tối ưu hơn, qua đó khẳng định giá trị và ý nghĩa của đề tài.

Vì những lý do trên, mục tiêu chính của đề tài là xây dựng một ứng dụng quản lý cửa hàng cafe F&B trên nền tảng WinForms sử dụng ngôn ngữ lập trình C#, nhằm hỗ trợ các chủ cửa hàng và quản lý vận hành hiệu quả các hoạt động kinh doanh. Ứng dụng tập trung vào việc tích hợp các chức năng cốt lõi, bao gồm quản lý sản phẩm (thêm, sửa, xóa thông tin sản phẩm, phân loại theo loại đồ uống và thực phẩm), quản lý tồn kho (theo dõi số lượng nhập/xuất, cảnh báo tự động về hàng tồn thấp), quản lý nhân viên (quản lý thông tin cá nhân, vai trò, lịch làm việc và hiệu suất), quản lý doanh thu (ghi nhận đơn hàng, hóa đơn và tính toán lợi nhuận), cùng với các báo cáo thống kê chi tiết (báo cáo doanh thu theo ngày/tháng/năm, phân tích tồn kho và hiệu suất bán hàng).

Bên cạnh đó, ứng dụng sẽ triển khai hai worker chạy nền định kỳ để nâng cao tính tự động hóa: (1) Worker kiểm soát tồn kho, định kỳ quét dữ liệu và gửi email cảnh báo đến quản lý nếu lượng hàng trong kho giảm xuống dưới mức có thể sử dụng cho 7 ngày (dựa trên dữ liệu tiêu thụ trung bình), giúp tránh tình trạng thiếu hụt hàng hóa và tối ưu hóa nguồn cung; (2) Worker cập nhật báo cáo doanh thu hàng tháng, tự động tổng hợp dữ liệu và gửi email báo cáo chi tiết đến chủ cửa hàng, hỗ trợ phân tích xu hướng kinh doanh kịp thời.

Qua đó, đề tài hướng tới việc tạo ra một hệ thống phần mềm toàn diện, dễ sử dụng, giúp giảm thiểu lỗi thủ công, tăng cường hiệu quả quản lý và hỗ trợ ra quyết định kinh doanh trong ngành F&B.

**BẢNG PHÂN CÔNG CÔNG VIỆC**

| **MSSV** | **Họ và tên** | **Nhiệm vụ** | **Đánh giá** |
| --- | --- | --- | --- |
| CT06N0112 | Cấy Triều Đông | - Xử lý nghiệp vụ quản lý nhà cung cấp, quản lý chi nhánh  - Nhập xuất hàng hóa  - Triển khai code nền Repository và UnitOfWork  - Viết báo cáo | Chưa hoàn thành nhiệm vụ |
| CT07N0111 | Mạch Tiến Duy | - Triển khai worker báo cáo  - Lập trình giao diện Window Form  - Nghiệp vụ quản lý sản phẩm  - Thiết kế Database | Hoàn thành chưa đạt đầy đủ yêu cầu |
| CT07N0162 | Lê Thanh Yên | - Nghiệp vụ báo cáo  - Nghiệp vụ quản lý nguyên liệu, Role base access control  - Xác thực và ủy quyền | Hoàn thành chưa đạt đầy đủ yêu cầu |

**MỤC LỤC NỘI DUNG**

[**DANH MỤC TỪ VIẾT TẮT VÀ DIỄN GIẢI** 3](#_Toc209538345)

[**MỤC LỤC HÌNH ẢNH** 5](#_Toc209538346)

[**MỤC LỤC BẢNG** 6](#_Toc209538347)

[**CHƯƠNG I: CƠ SỞ LÝ THUYẾT** 7](#_Toc209538348)

[**1.** **Khảo sát, phân tích** 7](#_Toc209538349)

[**1.1.** **Khảo sát hệ thống quản lý thông tin mẫu tại Highland Coffee** 7](#_Toc209538350)

[**1.2.** **Khảo sát hệ thống quản lý thông tin mẫu tại Highland Coffee** 9](#_Toc209538351)

[**1.3.** **Phân tích hệ thống** 10](#_Toc209538352)

[**1.3.1.** **Các Tác nhân** 11](#_Toc209538353)

[**1.3.2.** **Phân tích các usecase dựa trên chức năng** 12](#_Toc209538354)

[**CHƯƠNG II: PHÂN TÍCH THIẾT HỆ THỐNG** 19](#_Toc209538355)

[**1.** **Các đối tượng hệ thống quản lý** 19](#_Toc209538356)

[**1.1.** **Users (người dùng)** 19](#_Toc209538357)

[**1.2.** **Employees (nhân viên)** 20](#_Toc209538358)

[**1.3.** **Branches (Chi Nhánh)** 20](#_Toc209538359)

[**1.4.** **Products (Sản Phẩm)** 21](#_Toc209538360)

[**1.5.** **Ingredients (Nguyên Liệu)** 22](#_Toc209538361)

[**1.6.** **Suppliers (Nhà Cung Cấp)** 22](#_Toc209538362)

[**1.7.** **Customers (Khách Hàng)** 23](#_Toc209538363)

[**1.8.** **Orders (Đơn Hàng)** 23](#_Toc209538364)

[**1.9.** **Ingredient Purchase Orders (Đơn Mua Nguyên Liệu)** 24](#_Toc209538365)

[**1.10.** **Inventory (Tồn Kho)** 24](#_Toc209538366)

[**2.** **Thiết kế triển khai code** 26](#_Toc209538367)

[**2.1.** **Tổng quan về mô hình triển khai** 26](#_Toc209538368)

[**2.2.** **Tổ chức mã nguồn** 29](#_Toc209538369)

[**2.3.** **Công nghệ sử dụng** 31](#_Toc209538370)

[**CHƯƠNG III: THIẾT KẾ GIAO DIỆN HỆ THỐNG** 33](#_Toc209538371)

[**1.** **Đăng nhập** 33](#_Toc209538372)

[**2.** **Thanh điều hướng bên hông** 34](#_Toc209538373)

[**3.** **Trang chủ** 35](#_Toc209538374)

[**4.** **Giao diện Danh sách chung** 36](#_Toc209538375)

[**5.** **Nhân sự** 37](#_Toc209538376)

[**6.** **Tài khoản** 40](#_Toc209538377)

[**7.** **Vai trò và quyền** 42](#_Toc209538378)

[**8.** **Hàng hóa** 44](#_Toc209538379)

[**9.** **Nhà cung cấp** 47](#_Toc209538380)

[**10.** **Sản phẩm** 49](#_Toc209538381)

[**11.** **Hệ thống gửi email thông báo** 52](#_Toc209538382)

[**CHƯƠNG IV: TRIỂN KHAI MÃ NGUỒN VÀ THIẾT KẾ LOGIC** 58](#_Toc209538383)

[**1.** **Về tổng quan thiết kế mã nguồn** 58](#_Toc209538384)

[**2. Tổ chức mã nguồn trong Solution** 60](#_Toc209538385)

[**3. Luồng xử lý nghiệp vụ** 61](#_Toc209538386)

[**CHƯƠNG V: ĐÁNH GIÁ HỆ THỐNG** 62](#_Toc209538387)

[**1. Ưu điểm** 62](#_Toc209538388)

[**2. Hạn chế** 62](#_Toc209538389)

[**CHƯƠNG VI: HƯỚNG PHÁT TRIỂN** 63](#_Toc209538390)

[**KẾT LUẬN** 63](#_Toc209538391)

[**TÀI LIỆU THAM KHẢO** 65](#_Toc209538392)

# **DANH MỤC TỪ VIẾT TẮT VÀ DIỄN GIẢI**

| **Viết tắt** | **Thuật ngữ Tiếng Anh** | **Diễn giải Tiếng Việt** |
| --- | --- | --- |
| F&B | Food and Bẻverage | ngành công nghiệp thực phẩm và đồ uống |
| C# | Csharp | Ngôn ngữ lập trình hướng đối tượng C thăng |
| Winform | [Windows Forms](https://www.google.com/search?client=firefox-b-d&sca_esv=5b71cfa89013eb9f&sxsrf=AE3TifOMZ-NoLrPHY50_MGqHpw4Js_b9FA%3A1757928391370&q=Windows+Forms&sa=X&ved=2ahUKEwjgrLvKudqPAxU2bfUHHaNfFbYQxccNegQIKBAB&mstk=AUtExfAb7-mohm6IT2_YuFR4dX8cUmeHYiqssvMAxVTQssC-ZTFAAjo9ruCtTboSlg6Z5Xta_xuhNEPAzIRx9FRzJb33H5yyolFZnvPg0mfqwO7YrAwhL4LZAvDd2fOElmT9BxBxq2JyiVLUvZeUesT0iLMcRuTkkB1GsF_SztuuzWSFfjmht5JZw4pBFfLCSsLTUBRMeXEQtHFVeZXbIICwpR5ddrDlT2qRnUf9WMb68RCv6-1ZpbQLf5VaGJj_6vSpTwLyZpqw-hcn_1Y9UG_isHoo&csui=3) | Là một thư viện lớp đồ họa (GUI) của Microsoft dùng để phát triển ứng dụng desktop cho hệ điều hành Windows, hỗ trợ lập trình viên tạo giao diện người dùng (GUI) bằng cách kéo thả |
|  | Worker | Worker là một ứng dụng chạy nền liên tục, thường không có giao diện người dùng, được dùng để thực hiện các tác vụ lặp đi lặp lại, định kỳ hoặc xử lý dài hạn (background jobs). |
| POS | Point of sales | là giải pháp tổng hợp hỗ trợ việc bán hàng và quản lý cửa hàng hiệu quả. Hệ thống POS tích hợp nhiều tính năng như quản lý sản phẩm, thanh toán (tiền mặt, quẹt thẻ ATM, ví điện tử), in hóa đơn, quản lý khách hàng và theo dõi báo cáo doanh thu. |
| SWOT | Strengths, Weaknesses, Opportunities, Threats | là một **mô hình phân tích chiến lược** thường dùng trong kinh doanh, quản lý và cả nghiên cứu đề tài. S: điểm mạnh, W: điểm yếu, O: cơ hội, T: rủi ro. SWOT giúp nhìn tổng thể về hiện trạng, ưu điểm – nhược điểm. Xác định cơ hội phát triển và rủi ro tiềm ẩn. Đưa ra chiến lược phát triển hợp lý dựa trên việc phát huy điểm mạnh, khắc phục điểm yếu, tận dụng cơ hội và hạn chế thách thức. |
|  | Ecommerc | Thương mại điện tử |
| OOP | Object Oriented Programming | Lập trình hướng đối tượng |
| Async | Asynchronus | Bất đồng bộ |
| SRP | Single Responsibility Principle | là nguyên lý thứ nhất thuộc bộ nguyên lý SOLID trong thiết kế và phát triển phần mềm hướng đối tượng. |
| SoC | Separation of Concerns | Một nguyên lý quan trọng và cơ bản trong việc thiết kế và kiến trúc phần mềm. Ý tưởng của nguyên lý này rất đơn giản, đó là mỗi một object chỉ nên biết và thực thi công việc của chính nó. |
| RBAC | Role-based Access Control | Kiểm soát truy cập theo vai trò |
| CPM | Central Package Version Management | Quản lý phiên bản gói tập trung |
|  | Package | Gói, các thư viện được viết và đóng gói thành các phiên bản trong dotnet |
|  | Version | Phiên bản |
|  | Web ecommerce | Website thương mại điện tử |
|  | Layer architectured | Kiến trúc phân lớp |
|  | Monolith. | kiến trúc nguyên khối, hay liền khối là kiến trúc phần mềm tất cả trong một (all-in-one). |
|  | Build | Thực hiện biên diện mã nguồn sang kiểu file có thể chạy được |
|  | Project | Dự án, quản lý mã nguồn trong dotnet |
| MVVM | Model view view-model | Model - View - ViewModel, ở đó view (tức giao diện người dùng) sẽ được cập nhật bởi ViewModel và việc xử lý Logic hoặc trình bày dữ liệu sẽ do Model đảm nhận. Thường được ứng dụng trong các ứng dụng winform hay mobile app |
| UI/UX | User interface / User expirences | UI bao gồm mọi thứ mà người dùng tiếp xúc khi họ truy cập trang web hoặc ứng dụng. Tuy nhiên, trải nghiệm người dùng (UX) là chính. Nói cách khác, đây là cách người dùng tương tác với các thành phần giao diện tạo ra. |
| DTO | Data transfer object | Có thể xem là phiên bản thu gọn hoặc cụ thể hóa của các thực thể khi được truyền đi trong hệ thống nhằm phụ vụ các chức năng cụ thể. |

# **MỤC LỤC HÌNH ẢNH**

[Hình 1.Sơ đồ ERD rút gọn. 25](#_Toc209535310)

[Hình 2. Luồng đi dữ liệu của mô hình MVVM 27](#_Toc209535311)

[Hình 3. Cấu trúc của solution. 30](#_Toc209535312)

[Hình 4. Quản lý phiên bản gói tập trung 32](#_Toc209535313)

[Hình 5. Giao diện đăng nhập. 33](#_Toc209535314)

[Hình 6. Hiển thị thông báo lỗi thông qua toast message khi người dùng lỗi hoặc server lỗi 34](#_Toc209535315)

[Hình 7. Cho phép hiển thị mật khẩu để dễ dàng nhập hơn hoặc tắt đi để đảm bảo an toàn 34](#_Toc209535316)

[Hình 8. Chỉ hiệu đang tải giúp người dùng nhận biết tình trạng ứng dụng đồng thời ngăn người dùng tạo thêm sự kiện mới gây xun đột 34](#_Toc209535317)

[Hình 9. Thanh điều hướng. 35](#_Toc209535318)

[Hình 10. Các thành pần có thể thực hiện đóng mở 35](#_Toc209535319)

[Hình 11. Giao diện trang chủ. 36](#_Toc209535320)

[Hình 12. Mô tả giao diện hiển thị danh sách. 37](#_Toc209535321)

[Hình 13. Danh sách nhân sự. 38](#_Toc209535322)

[Hình 14. Màn hình thêm/chỉnh sửa nhân viên. 38](#_Toc209535323)

[Hình 15. Thông tin về tên tài khoản và vai trò của người dùng trong hệ thống. 39](#_Toc209535324)

[Hình 16. Danh sách tài khoản. 40](#_Toc209535325)

[Hình 17. Giao diện thêm / sửa người dùng. 41](#_Toc209535326)

[Hình 18. Giao diện cho phép thực hiện gán vai trò cho người dùng. 41](#_Toc209535327)

[Hình 19. Danh sách vai trò và quyền. 43](#_Toc209535328)

[Hình 20. Trang thêm / sửa vai trò 43](#_Toc209535329)

[Hình 21. Trang thêm sửa quyền hạn 44](#_Toc209535330)

[Hình 22. Trang gán quyền hạn cho vai trò 44](#_Toc209535331)

[Hình 23. Danh sách hàng hóa. 46](#_Toc209535332)

[Hình 24. Thêm nguyên liệu 46](#_Toc209535333)

[Hình 25. Chi tiết nguyên liệu và cập nhật thông tin 47](#_Toc209535334)

[Hình 26. Danh sách nhà cung cấp. 48](#_Toc209535335)

[Hình 27. Chi tiết thông tin về nhà cung cấp (có thể thêm sửa hoặc xóa). 49](#_Toc209535336)

[Hình 28. Giao diện danh sách thông tin sản phẩm. 50](#_Toc209535337)

[**Hình 29. Giao diện chi tiết sản phẩm.** 50](#_Toc209535338)

[Hình 30. Danh sách hình ảnh của một sản phẩm. 52](#_Toc209535339)

[Hình 31. Danh sách chi tiết càc công thức liên qua 53](#_Toc209535340)

[Hình 32. Giao diện email thông báo hết hàng. 54](#_Toc209535341)

[Hình 33. File csv gửi đính kèm thông báo 54](#_Toc209535342)

[Hình 34. Cấu trúc dự án. 60](#_Toc209535343)

# **CHƯƠNG I: CƠ SỞ LÝ THUYẾT**

## **Khảo sát, phân tích**

## **Khảo sát hệ thống quản lý thông tin mẫu tại Highland Coffee**

Để xây dựng nền tảng lý thuyết vững chắc cho đề tài, nhóm đã thực hiện khảo sát chi tiết về hệ thống quản lý cửa hàng cafe F&B, tập trung vào đối tượng thực tế là Highlands Coffee – một trong những cà phê hàng đầu tại Việt Nam với hơn 500 cửa hàng trên toàn quốc (dữ liệu cập nhật đến năm 2025). Khảo sát được tiến hành dựa trên phương pháp kết hợp phân tích tài liệu thứ cấp (báo cáo case study, tài liệu học thuật từ các nguồn như ResearchGate *(Nguyen, 2024)*, Studocu *(P. H. Trang, 2021)*, Scribe và một số bài báo liên quan) và tổng hợp dữ liệu từ các nguồn công khai về hoạt động của Highlands. Mục tiêu là xác định các đối tượng tương tác chính trong cửa hàng cafe F&B, bao gồm hàng hóa/sản phẩm bán ra, tồn kho, tài chính, người dùng hệ thống (chủ cửa hàng, quản lý, nhân viên) và nhân viên chi nhánh. Thời gian khảo sát diễn ra từ tháng 8 đến tháng 9 năm 2025, với nguồn tài liệu liên quan đến quản lý hoạt động (operations management) của Highlands.

Phương pháp khảo sát:

* Thu thập dữ liệu thứ cấp: Phân tích các case study như "Highlands Coffee: Brewing Success in Vietnam" (B2Bnn, 2025), báo cáo "A Strategic Analysis of Highlands Coffee's Success" (ResearchGate, 2025), và các tài liệu nội bộ từ Studocu/Scribd về cung ứng và quản lý nhân sự.
* Phạm vi đối tượng: Tập trung vào 5 nhóm chính: (1) Hàng hóa và sản phẩm bán ra; (2) Tồn kho và cung ứng; (3) Tài chính và doanh thu; (4) Người dùng hệ thống (quản trị viên/quản lý/nhân viên); (5) Nhân viên chi nhánh.
* Công cụ phân tích: Sử dụng bảng so sánh và sơ đồ để tổng hợp dữ liệu, nhằm xác định nhu cầu chức năng và phi chức năng của hệ thống quản lý.

| **Đối tượng** | **Mô tả chi tiết từ khảo sát Highlands Coffee** | **Nhu cầu quản lý cụ thể** | **Thách thức hiện tại** |
| --- | --- | --- | --- |
| Hàng hóa và sản phẩm bán ra | Highlands cung cấp đa dạng sản phẩm: đồ uống truyền thống (cà phê sữa đá, trà sen vàng, trà xanh đá xay) và hiện đại (espresso, latte, cappuccino); sản phẩm đóng gói RTD (ready-to-drink) iced coffee xuất khẩu sang 17 quốc gia châu Á và Mỹ. Menu thay đổi theo mùa, với nguồn nguyên liệu từ ~1.000 nông trại tại Lâm Đồng và Đắk Lắk. Quy trình bán hàng bao gồm POS (Point-of-Sale) system cho order nhanh và upsell techniques (gợi ý sản phẩm kèm theo). | Theo dõi danh mục sản phẩm, cập nhật giá, phân loại (đồ uống/thực phẩm), tích hợp upsell tự động. | Menu đa dạng dẫn đến khó khăn trong theo dõi biến động giá nguyên liệu; thiếu tích hợp dữ liệu thời gian thực cho khuyến mãi. |
| Tồn kho và cung ứng | cung ứng khép kín: từ thu mua hạt cà phê trực tiếp từ nông dân → xử lý, rang xay tại nhà máy (ví dụ: nhà máy rang xay mới tại Bà Rịa - Vũng Tàu trị giá 19,3 triệu USD, khai trương 2025) → phân phối đến cửa hàng. Sử dụng hệ thống quản lý tồn kho cơ bản để theo dõi nhập/xuất, nhưng chủ yếu thủ công kết hợp phần mềm đơn giản. Trong đại dịch COVID-19 (2021), gặp khó khăn trong phân bổ tồn kho cho kênh online. | Dự báo nhu cầu tồn kho dựa trên dữ liệu bán hàng, cảnh báo tự động khi tồn kho dưới mức an toàn (ví dụ: dưới 7 ngày tiêu thụ), tích hợp với nhà cung cấp. | Phân bổ tồn kho không đồng đều giữa chi nhánh; lãng phí do hết hạn (đặc biệt đồ uống tươi); thiếu tự động hóa dẫn đến lỗi nhập liệu. |
| Tài chính và doanh thu | Doanh thu chính từ bán lẻ tại cửa hàng (80% tổng doanh thu), kết hợp online qua app/website và đối tác như Boxme (logistics). Sử dụng POS để ghi nhận hóa đơn, tính lợi nhuận thời gian thực. Năm 2022, doanh thu tăng 30% nhờ chuyển đổi số, nhưng báo cáo tài chính hàng tháng chủ yếu thủ công qua Excel. | Tính toán doanh thu theo ngày/tháng/chi nhánh, báo cáo lợi nhuận, phân tích xu hướng (ví dụ: sản phẩm bán chạy). | Báo cáo chậm trễ, khó tích hợp dữ liệu từ nhiều chi nhánh; rủi ro gian lận trong thanh toán (tiền mặt vs. thẻ). |
| Người dùng hệ thống (chủ cửa hàng, quản lý, nhân viên) | Chủ cửa hàng/quản lý: Quyết định chiến lược, xem báo cáo tổng hợp. Nhân viên: Xử lý order, cập nhật tồn kho hàng ngày. Hệ thống hiện tại: POS cho nhân viên bán hàng, dashboard cơ bản cho quản lý. Đào tạo người dùng tập trung vào giao diện thân thiện. | Phân quyền truy cập (admin cho chủ, read-only cho nhân viên), giao diện đa cấp (WinForms cho desktop tại cửa hàng). | Giao diện POS cũ kỹ, không thân thiện với nhân viên mới; thiếu đào tạo tích hợp hệ thống dẫn đến lỗi sử dụng. |
| Nhân viên chi nhánh | Khoảng 5-10 nhân viên/chi nhánh, vai trò: barista, thu ngân, phục vụ. Đào tạo nhấn mạnh kiến thức cà phê, dịch vụ khách hàng (attentive service), và phản hồi nhanh. Tỷ lệ nghỉ việc cao (ngành F&B trung bình 40%/năm). Quản lý lịch làm việc thủ công, thiếu hệ thống theo dõi hiệu suất. | Quản lý lịch làm việc, đánh giá hiệu suất, đào tạo trực tuyến qua hệ thống. | Thiếu nhân sự giờ cao điểm; đào tạo không đồng đều giữa chi nhánh; quản lý đa địa điểm khó khăn (hơn 500 cửa hàng). |

Từ khảo sát, có thể thấy Highlands Coffee đã áp dụng một số công nghệ cơ bản như POS và app di động, nhưng hệ thống tổng thể còn rời rạc, dẫn đến hiệu quả quản lý chưa tối ưu.

## **Khảo sát hệ thống quản lý thông tin mẫu tại Highland Coffee**

Dựa trên dữ liệu khảo sát, phân tích hệ thống được thực hiện theo mô hình SWOT (Strengths, Weaknesses, Opportunities, Threats) để xác định các yêu cầu chức năng và phi chức năng cho ứng dụng quản lý.

- Điểm mạnh (Strengths): cung ứng mạnh mẽ với nguồn nguyên liệu địa phương đảm bảo chất lượng; POS hỗ trợ bán hàng nhanh; đào tạo nhân viên tập trung vào dịch vụ khách hàng, giúp Highlands đạt doanh thu cao (tăng 30% sau đại dịch).

- Điểm yếu (Weaknesses): Tồn kho thủ công dẫn đến lãng phí (ước tính 5-10% hàng hóa hết hạn); quy trình order và thanh toán chậm (thời gian trung bình 3-5 phút/order, theo báo cáo CourseHero 2025); báo cáo tài chính không thời gian thực, khó hỗ trợ quyết định nhanh cho chủ cửa hàng.

- Cơ hội (Opportunities): Tích hợp tự động hóa (như worker gửi email cảnh báo tồn kho) để giảm lỗi; mở rộng online sales (Highlands đã hợp tác Boxme, có thể tích hợp API cho theo dõi doanh thu); đào tạo nhân viên qua hệ thống số hóa để giảm tỷ lệ nghỉ việc.

- Thách thức (Threats): Cạnh tranh gay gắt từ Starbucks, The Coffee House; biến động giá nguyên liệu (tăng 15% năm 2025 do lạm phát); rủi ro an ninh dữ liệu từ hệ thống POS cũ.

**Yêu cầu chức năng chính từ phân tích:**

- Quản lý thể loại/sản phẩm/nhà cung cấp/ca làm: CRUD (Create, Read, Update, Delete) cho menu, hiệu suất của nhà cung cấp.

- Quản lý tồn kho: Theo dõi nhập/xuất từ kho tới chi nhánh và nhập mua hàng, dự báo dựa trên dữ liệu lịch sử bán hàng.

- Báo cáo thống kê doanh thu: Tính toán hóa đơn, báo cáo thống kê (biểu đồ doanh thu theo chi nhánh).

- Quản lý nhân sự: thông tin nhân sự.

- Tính năng nâng cao: Worker định kỳ gửi email (tồn kho thấp, báo cáo tháng).

**Một số chức năng ứng dụng kèm theo:**

- Xác thực thông tin người dùng.

- Phân quyền truy cập theo quyền hạn và vai trò.

## **Phân tích hệ thống**

Phân tích hệ thống được thực hiện dựa trên các yêu cầu chức năng đã xác định từ khảo sát và phân tích ở Chương 1.2, tập trung vào việc mô hình hóa các tương tác giữa người dùng và hệ thống. Ứng dụng quản lý cửa hàng cafe F&B được thiết kế để hỗ trợ quản lý nội bộ, không tương tác trực tiếp với khách hàng (phần này sẽ được xử lý bởi hệ thống web ecommerce riêng). Phân tích sử dụng mô hình UML (Unified Modeling Language), bao gồm Use Case Diagram để xác định các Tác nhân và use case chính.

Dưới đây là phân tích chi tiết về các Tác nhân (các thực thể tương tác với hệ thống) và Use Case (các trường hợp sử dụng). Các Tác nhân được xác định dựa trên đối tượng người dùng từ phần 1.1 (quản trị viên hệ thống thông tin, quản lý, nhân viên kế toán, nhân viên bán hàng, nhân viên kho v.v.), kết hợp với các thành phần tự động của hệ thống. Mỗi use case được đặc tả chi tiết theo định dạng chuẩn: Tên use case, Mô tả, Tác nhân liên quan, điều kiện trước, điều kiện sau, luồng chính, luồng thay thế, và luồng ngoại lệ.

### **Các Tác nhân**

Về vấn đề phân quyền cơ bản ta có một số vai trò cố định sau (không thể thay đổi):

1. **Quản trị viên (Administrator - Role: ADMIN):** Người quản trị toàn hệ thống, có quyền sửa/xóa toàn cục (UPDATE/DELETE cho product, ingredient, category, financial reports) và full CRUD cho tất cả chức năng. Họ xử lý cấu hình cao cấp, delegate permission, và hỗ trợ quyết định kinh doanh toàn .
2. **Quản lý (Manager - Role: MANAGER):** Quản lý hoạt động chi nhánh, có quyền sửa/xóa chỉ nội bộ chi nhánh (UPDATE/DELETE cho chi phí chi nhánh, kho hàng hóa/local inventory tại chi nhánh). Họ có quyền đọc toàn cục (READ cho products, reports, branches) và CREATE/READ/UPDATE orders/inventory trong chi nhánh.
3. **Giám sát viên (Supervisor - Role: SUPERVISOR):** Giám sát hoạt động ca làm việc, chỉ có quyền đọc (READ) dữ liệu như products, orders, inventory, financial reports, và branches. Họ theo dõi mà không chỉnh sửa.
4. **Nhân viên kho (Warehouse Staff - Role: WAREHOUSE\_STAFF):** Quản lý kho chi nhánh, chỉ có quyền đọc (READ) products và inventory (có thể READ/UPDATE local stock trong chi nhánh dưới sự giám sát, nhưng không xóa toàn cục).
5. **Thu ngân (Cashier - Role: CASHIER):** Xử lý thanh toán tại quầy, chỉ có quyền đọc (READ) products và CREATE/READ/UPDATE orders trong ca làm việc (không xóa hoặc sửa toàn cục).
6. **Nhân viên (Employee - Role: EMPLOYEE):** Nhân viên pha chế/phục vụ, chỉ có quyền đọc (READ) products và CREATE/READ orders cơ bản (không sửa/xóa bất kỳ dữ liệu nào).
7. **Hệ thống Worker (System Worker):** Thành phần tự động chạy nền, xử lý nhiệm vụ định kỳ như kiểm tra stock ingredient (gửi cảnh báo nếu thấp ở chi nhánh) và gửi báo cáo doanh thu hàng tháng (chỉ gửi, không chỉnh sửa). Không phụ thuộc Tác nhân người dùng.

Ngoài ra thì còn một số user khác để định danh người dùng hệ thống ở phía webclient tuy nhiên em sẽ không đề cập ở đây. Các Tác nhân này tương tác với hệ thống qua giao diện WinForms, với phân quyền dựa trên vai trò. Tuy nhiên vì thời gian có hạn nên trong dự án này chúng em sẽ rút gọn lại còn Actor chính là Quản trị viên, Quản Lý và nhân viên.

### **Phân tích các usecase dựa trên chức năng**

*Use Case UC\_LOGIN: Đăng nhập hệ thống*

* **Mô tả:** Cho phép người xác thực truy để có thể truy cập vài các tài nguyên của hệ thống tùy theo vai trò cảu người dùng.
* **Tác nhân liên quan:** Tất cả người dùng được gán vai trong 6 vai trò được nêu trên
* **Điều kiện trước:** Người dùng mở ứng dụng, và thời gian phiên trước đó đã hết.
* **Điều kiện sau:** Người dùng được xác thực thành công thì chuyển tới Form chính của ứng dụng,
* **Luồng cơ bản:**

1. Người dùng mở ứng dụng.
2. Thực hiện nhập thông tin mật khẩu và tài khoản vào ô nhập thông tin.
3. Người dùng ấn nút đăng nhập.
4. Người dùng được xác thực thành công thì chuyển tới Form chính của ứng dụng

* **Luồng thay thế:**

1. Người dùng nhập sai thôngtài khoản hoặc mật khẩu.
2. Thông báo lỗi ra cho người dùng và yêu cầu nhập lại.

* **Luồng ngoại lệ:** Thông báo lỗi hệ thống yêu cầu người dùng nhập lại sau

*Use Case UC\_QLSP:*

* **Mô tả**: Cho phép quản trị viên thưc hiện thêm sửa xóa các sản phẩm trên hệ thống.
* **Tác nhân liên quan**: Quản trị viên
* **Điều kiện trước**: Người dùng đăng nhập thành công với vai trò là quản trị viên.
* **Điều kiện sau**: Thao tác lên đối tượng được cập nhật vào cơ sở dữ liệu thành công.
* **Luồng cơ bản**:

1. Người dùng chọn vào trang sản phẩm trên sidebar.
2. Người dùng coi và thực hiện nhất đúp chuột hoặc ấn vào nút chi tiết để hiển thị chi tiết thông tin sản phẩm.
3. Người dùng thay đổi thông tin trong phần xem chi tiết sản phẩm.
4. Người dùng thực hiện ấn cập nhật.
5. Dữ liệu được hệ thống xác thực
6. Xác thực thành công thì cập nhật vào cơ sở dữ liệu
7. Hiển thị thông báo tới cho người dùng.

* **Luồng thay thế**:

3.b. Người dùng không thực hiện thay đổi và bấm hủy 🡪 thoát khỏi giao diện chi tiết.

3.c. Người dùng không cập nhật thông tin sản phẩm, bấm vào tabs công thức.

4.c. Người dùng thao tác thay đổi thông tin trong phần công thức.

5.c. Hệ thống xác thực dữ liệu

6.c. Xác thực thành công 🡪 Lưu vào cơ sở dữ liệu.

7.c. Hiển thị thông báo kết quả tới người dùng.

* **Luồng ngoại lệ:** thông báo lỗi và yêu cầu người dùng thử lại sau.

*Use Case UC\_QLNL: Quản lý nguyên liệu*

* **Mô tả**: Cho phép quản trị viên thực hiện thêm sửa xóa các sản phẩm trên hệ thống.
* **Tác nhân liên quan**: Quản trị viên
* **Điều kiện trước**: Người dùng đăng nhập thành công với vai trò là quản trị viên.
* **Điều kiện sau**: Thao tác lên đối tượng được cập nhật vào cơ sở dữ liệu thành công.
* **Luồng cơ bản**:

1. Người dùng chọn vào trang sản phẩm trên sidebar.
2. Người dùng coi và thực hiện nhất đúp chuột hoặc ấn vào nút chi tiết để hiển thị chi tiết thông tin hàng hóa.
3. Người dùng thay đổi thông tin trong phần xem chi tiết nguyên liệu.
4. Người dùng thực hiện ấn cập nhật.
5. Dữ liệu được hệ thống xác thực
6. Xác thực thành công thì cập nhật vào cơ sở dữ liệu
7. Hiển thị thông báo tới cho người dùng.

* **Luồng thay thế**:

3.b. Người dùng không thực hiện thay đổi và bấm Hủy 🡪 thoát khỏi giao diện chi tiết.

* **Luồng ngoại lệ:** thông báo lỗi và yêu cầu người dùng thử lại sau.

Các Use Case quản lý chung còn lại như Quản lý nhân sự, nhà cung cấp, người dùng, nhân sự tương tự đối

*Use Case UC\_QLNV: Quản lý Nhân viên*

* **Mô tả**: Cho phép quản trị viên thực hiện thêm, sửa, xóa nhân viên toàn cục; quản lý chỉ CRUD nội bộ chi nhánh. Bao gồm tabs cấp tài khoản, thông tin lương, bảng lương.
* **Tác nhân liên quan:** Quản trị viên (full CRUD toàn cục), Quản lý (CRUD nội bộ chi nhánh), Giám sát viên/Nhân viên kho/Thu ngân/Nhân viên (READ only).
* **Điều kiện trước**: Người dùng đăng nhập thành công với vai trò phù hợp; cơ sở dữ liệu nhân viên tồn tại.
* **Điều kiện sau:** Thao tác lên nhân viên được cập nhật vào cơ sở dữ liệu thành công (giới hạn theo chi nhánh nếu Quản lý).
* **Luồng cơ bản**:
  1. Người dùng chọn vào trang nhân viên trên sidebar.
  2. Người dùng có thể xem, tìm kiếm đối tượng cần thao tác và thực hiện nhấp đúp chuột hoặc ấn vào nút chi tiết để hiển thị chi tiết thông tin nhân viên (hiển thị dưới dạng 4 tabs).
  3. Trong tabs đầu (Thông tin cơ bản): Người dùng thay đổi thông tin (tên, vai trò, lịch làm việc) – scoped by branch cho Quản lý.
  4. Người dùng thực hiện ấn cập nhật.
  5. Dữ liệu được hệ thống xác thực.
  6. Xác thực thành công thì cập nhật vào cơ sở dữ liệu.
  7. Hiển thị thông báo tới cho người dùng.
* **Luồng thay thế**: 3.b. Người dùng không thực hiện thay đổi và bấm hủy → thoát khỏi giao diện chi tiết. 3.c. Người dùng bấm vào tabs Cấp tài khoản: Ấn vào ô tài khoản, nhập thông tin đăng nhập (username, password) – chỉ Quản trị viên/Quản lý nội bộ; ấn lưu, hệ thống xác thực (kiểm tra trùng lặp). 4.c. Xác thực thành công → Lưu; không thì báo lỗi. 3.d. Người dùng bấm vào tabs Thông tin lương: Thêm/sửa/xóa danh sách (ngày hiệu lực, lương cơ bản, loại lương, phụ cấp, thưởng, khấu trừ, thuế %) – scoped for Quản lý. 4.d. Hệ thống xác thực dữ liệu. 5.d. Xác thực thành công → Lưu. 3.e. Người dùng bấm vào tabs Bảng lương: Xem/thêm/sửa (tháng năm, giờ làm, lương cơ bản, phụ cấp, thưởng, khấu trừ, tổng lương) – scoped. 4.e. Hệ thống xác thực và tính toán tổng. 5.e. Lưu và hiển thị.
* **Luồng ngoại lệ**: Thông báo lỗi (quyền không đủ, dữ liệu không hợp lệ) và yêu cầu thử lại.

*Use Case UC\_QLNCC: Quản lý Nhà cung cấp*

* **Mô tả**: Cho phép quản trị viên thực hiện thêm, sửa, xóa nhà cung cấp toàn cục; quản lý chỉ READ hoặc UPDATE nội bộ (hợp đồng chi nhánh).
* **Tác nhân liên quan**: Quản trị viên (full CRUD toàn cục), Quản lý (READ toàn cục, UPDATE nội bộ), Giám sát viên/Nhân viên kho (READ only).
* **Điều kiện trước**: Người dùng đăng nhập thành công với vai trò phù hợp; cơ sở dữ liệu nhà cung cấp tồn tại.
* **Điều kiện sau**: Thao tác lên nhà cung cấp được cập nhật vào cơ sở dữ liệu thành công (scoped).
* **Luồng cơ bản**:
  1. Người dùng chọn vào trang nhà cung cấp trên sidebar.
  2. Người dùng có thể xem, tìm kiếm đối tượng cần thao tác và thực hiện nhấp đúp chuột hoặc ấn vào nút chi tiết để hiển thị chi tiết thông tin nhà cung cấp.
  3. Người dùng thay đổi thông tin (tên, địa chỉ, hợp đồng) – scoped for Quản lý.
  4. Người dùng thực hiện ấn cập nhật.
  5. Dữ liệu được hệ thống xác thực.
  6. Xác thực thành công thì cập nhật vào cơ sở dữ liệu.
  7. Hiển thị thông báo tới cho người dùng.
* **Luồng thay thế**: 3.b. Người dùng không thực hiện thay đổi và bấm hủy → thoát khỏi giao diện chi tiết.
* **Luồng ngoại lệ**: Thông báo lỗi và yêu cầu thử lại.

*Use Case UC\_QLND: Quản lý Người dùng (Tài khoản)*

* **Mô tả**: Cho phép quản trị viên thực hiện thêm, sửa, xóa người dùng toàn cục; quản lý chỉ CRUD nội bộ chi nhánh. Bao gồm 3 tabs: danh sách, thêm mới, gán/gỡ vai trò.
* **Tác nhân liên quan**: Quản trị viên (full CRUD toàn cục), Quản lý (CRUD nội bộ), Giám sát viên/Nhân viên kho/Thu ngân/Nhân viên (READ only).
* **Điều kiện trước**: Người dùng đăng nhập thành công với vai trò phù hợp; cơ sở dữ liệu người dùng tồn tại.
* **Điều kiện sau**: Thao tác lên người dùng được cập nhật vào cơ sở dữ liệu thành công (scoped).
* **Luồng cơ bản**:
  1. Người dùng chọn vào trang người dùng trên sidebar (hiển thị dưới dạng 3 tabs).
  2. Trong tabs Danh sách người dùng: Xem, tìm kiếm, nhấp đúp hoặc nút chi tiết để hiển thị chi tiết.
  3. Người dùng thay đổi thông tin (username, password) – scoped.
  4. Ấn cập nhật.
  5. Hệ thống xác thực.
  6. Cập nhật cơ sở dữ liệu.
  7. Hiển thị thông báo.
* **Luồng thay thế**: 2.b. Bấm tabs Thêm người dùng mới: Nhập thông tin mới (thay vì dialog) – scoped. 3.b. Ấn lưu, xác thực và lưu. 2.c. Bấm tabs Gán vai trò: Chọn người dùng, gán/gỡ role (e.g., MANAGER) – chỉ Quản trị viên full, Quản lý nội bộ. 3.c. Hệ thống cập nhật role\_permissions. 4.c. Hiển thị kết quả.
* **Luồng ngoại lệ**: Thông báo lỗi (trùng username, quyền không đủ).

*Use Case UC\_PQ:*

* **Mô tả**: Cho phép quản trị viên tạo/sửa quyền và gán quyền cho vai trò mong muốn.
* **Tác nhân liên quan**: Quản trị viên (full CRUD),
* **Điều kiện trước**: Đăng nhập phù hợp; Kết nối thành công với cơ sở dữ liệu
* **Điều kiện sau**: Cập nhật thành công.
* **Luồng cơ bản**: Tương tự UC\_QLNL, với thay đổi thông tin chi nhánh (địa chỉ, manager).
* **Luồng thay thế**: Không thay đổi, hủy.
* **Luồng ngoại lệ**: Lỗi quyền hoặc dữ liệu.

*Use Case UC\_QLKHO: Quản lý Kho (Stock Ingredient)*

* **Mô tả**: Cho phép quản trị viên full CRUD toàn cục; quản lý/Nhân viên kho UPDATE/DELETE nội bộ (transfer, order/mua stock, chi phí kho).
* **Tác nhân liên quan**: Quản trị viên (full), Quản lý/Nhân viên kho (UPDATE nội bộ), Giám sát viên (READ).
* **Điều kiện trước**: Đăng nhập phù hợp.
* **Điều kiện sau**: Cập nhật thành công, worker kiểm tra nếu thấp.
* **Luồng cơ bản**: Tương tự UC\_QLNL, với thêm transfer/order/mua (chọn supplier, cập nhật chi phí).
* **Luồng thay thế**: Transfer giữa branch (scoped), order mới.
* **Luồng ngoại lệ**: Stock âm, lỗi.

*Use Case UC\_QLDH: Quản lý Đơn hàng*

* **Mô tả**: Cho phép thu ngân/nhân viên CREATE/READ/UPDATE orders nội bộ; quản trị viên full.
* **Tác nhân liên quan**: Quản trị viên (full), Quản lý (CRUD nội bộ), Thu ngân (CREATE/READ/UPDATE), Nhân viên (CREATE/READ), Giám sát viên (READ).
* **Điều kiện trước**: Đăng nhập phù hợp; tồn kho đủ.
* **Điều kiện sau**: Order lưu, tồn kho cập nhật.
* **Luồng cơ bản**: Chọn sản phẩm, tính giá, xác nhận thanh toán, lưu.
* **Luồng thay thế**: Upsell gợi ý, hủy order.
* **Luồng ngoại lệ**: Hết hàng, lỗi thanh toán.

*Use Case UC\_QLRole: Quản lý Role và Permission*

* **Mô tả**: Chỉ quản trị viên full CRUD/delegate toàn cục.
* **Tác nhân liên quan**: Quản trị viên.
* **Điều kiện trước**: Đăng nhập ADMIN.
* **Điều kiện sau**: Cập nhật role\_permissions.
* **Luồng cơ bản**: Tương tự mẫu, với delegate permission.
* **Luồng thay thế**: Xem permissions.
* **Luồng ngoại lệ**: Trùng lặp.

*Use Case UC\_XBC: Xem Báo cáo*

* **Mô tả**: Xem báo cáo doanh thu, lợi nhuận, biểu đồ tình trạng, hàng sắp hết; worker gửi định kỳ. Quản trị viên full (UPDATE reports nếu cần); quản lý/supervisor READ nội bộ.
* **Tác nhân liên quan**: Quản trị viên (full), Quản lý/Giám sát viên (READ nội bộ), Nhân viên kho (READ hàng sắp hết).
* **Điều kiện trước**: Đăng nhập phù hợp; dữ liệu tồn tại.
* **Điều kiện sau**: Báo cáo hiển thị/export.
* **Luồng cơ bản**:
  1. Người dùng ấn vào trang chủ trên sidebar.
  2. Hệ thống hiển thị dashboard: Biểu đồ tình trạng (doanh thu theo thời gian chọn), thông tin hàng hóa sắp hết (dựa trên worker).
  3. Chọn thời gian (ngày/tháng/năm) để xem báo cáo doanh thu/lợi nhuận.
  4. Hệ thống tổng hợp dữ liệu (LINQ queries).
  5. Hiển thị biểu đồ và chi tiết (Chart control).
  6. Nếu Quản trị viên, cho phép UPDATE dữ liệu báo cáo (sửa chi phí).
  7. Hiển thị thông báo hoặc export.
* **Luồng thay thế**: Xem hàng sắp hết → Chuyển sang UC\_QLKHO; export PDF/email (worker tự động hàng tháng).
* **Luồng ngoại lệ**: Không có dữ liệu, hiển thị thông báo; lỗi quyền, từ chối UPDATE.

# **CHƯƠNG II: PHÂN TÍCH THIẾT HỆ THỐNG**

## **Các đối tượng hệ thống quản lý**

Trong hệ thống quản lý quán cà phê, các đối tượng chính được thiết kế để mô hình hóa các thành phần cốt lõi của quy trình kinh doanh, bao gồm quản lý người dùng, nhân sự, chi nhánh, sản phẩm, nguyên liệu, nhà cung cấp, khách hàng, đơn hàng và tồn kho. Các đối tượng này được xây dựng dựa trên mô hình dữ liệu quan hệ (Relational Data Model), hỗ trợ các tính năng như xác thực, phân quyền, theo dõi tài chính và quản lý cung ứng. Dưới đây là mô tả chi tiết về từng đối tượng chính, bao gồm thuộc tính, mối quan hệ, vai trò và các chi tiết bổ sung.

## **Users (người dùng)**

Đối tượng Users đại diện cho các tài khoản người dùng trong hệ thống, bao gồm cả nhân viên và khách hàng, nhằm hỗ trợ xác thực và phân quyền dựa trên cơ chế Role-Based Access Control (RBAC).

**Thuộc tính chính:**

* id: Khóa chính.
* employee\_id: Liên kết với nhân viên theo quan hệ 1 – 1.
* username: tên tài khoản người dùng.
* address: Địa chỉ (đã mã hóa).
* password: Mật khẩu đã được hash + salt
* role\_id: Khóa ngoại liên kết với vai trò.
* is\_active: Trạng thái kích hoạt tài khoản.

**Mối quan hệ:** Liên kết với bảng roles (vai trò), tokens (cho JWT authentication), social\_accounts (đăng nhập xã hội), và customers (nếu là khách hàng).

**Vai trò:** Quản lý quy trình đăng ký, đăng nhập, quên mật khẩu và phân quyền. Hệ thống đảm bảo bảo mật thông qua hash mật khẩu với salt, mã hóa dữ liệu nhạy cảm như địa chỉ, và các biện pháp chống tấn công như phát hiện nhiều lần đăng nhập thất bại từ cùng một địa chỉ IP.

**Chi tiết bổ sung:** Hệ thống lưu trữ lịch sử mật khẩu ẩn (không hiển thị trong schema nhưng được triển khai theo yêu cầu), hạn sử dụng mật khẩu, và cơ chế khóa tạm thời tài khoản để tăng cường an ninh.

## **Employees (nhân viên)**

Đối tượng Employees quản lý thông tin nhân viên làm việc tại các chi nhánh, hỗ trợ theo dõi nhân sự và tính toán lương.

**Thuộc tính chính:**

* id: Khóa chính.
* branch\_id: Khóa ngoại liên kết với chi nhánh.
* full\_name: Họ và tên.
* phone: Số điện thoại.
* email: Địa chỉ email.
* position: Chức vụ.
* hire\_date: Ngày tuyển dụng.
* status: Trạng thái (hoạt động/nghỉ việc).

**Mối quan hệ:** Liên kết với branches (chi nhánh), users (tài khoản), payrolls (lương), employee\_shifts (ca làm việc), và các bảng liên quan đến kho hoặc đơn hàng.

**Vai trò:** Theo dõi nhân sự, tính lương hàng tháng (bao gồm lương cơ bản, phụ cấp, thưởng, phạt, thuế và khấu trừ), và phân công công việc. Hệ thống hỗ trợ tự động hóa quy trình payroll dựa trên dữ liệu ca làm và hiệu suất.

**Chi tiết bổ sung:** Sử dụng view v\_employee\_payroll để tóm tắt thông tin lương. Đối tượng này còn liên kết với inventory\_movements (di chuyển kho) để theo dõi trách nhiệm của nhân viên trong các hoạt động kho hàng.

## **Branches (Chi Nhánh)**

Đối tượng Branches đại diện cho các địa điểm kinh doanh (chi nhánh cửa hàng), hỗ trợ phân tán hoạt động kinh doanh đa chi nhánh.

**Thuộc tính chính:**

* id: Khóa chính.
* name: Tên chi nhánh.
* address: Địa chỉ.
* phone: Số điện thoại.
* manager: Quản lý chi nhánh (liên kết với employee\_id).

**Mối quan hệ:** Liên kết với employees (nhân viên tại chi nhánh), orders (đơn hàng), branch\_ingredient\_inventory (tồn kho nguyên liệu), branch\_expenses (chi phí), và các bảng tóm tắt như sales\_summary.

**Vai trò:** Quản lý kho riêng biệt cho từng chi nhánh, theo dõi hiệu suất bán hàng và chi phí. Hệ thống hỗ trợ báo cáo tổng hợp theo chi nhánh để đánh giá hiệu quả hoạt động.

**Chi tiết bổ sung:** Hỗ trợ multi-branch operations với ngưỡng tồn kho riêng (inventory thresholds) để cảnh báo và duy trì mức tồn kho an toàn.

## **Products (Sản Phẩm)**

**Mô tả:** Đối tượng Products quản lý các mặt hàng bán lẻ, bao gồm sản phẩm làm sẵn hoặc sản xuất từ nguyên liệu.

**Thuộc tính chính:**

* id: Khóa chính.
* name: Tên sản phẩm.
* description: Mô tả.
* price: Giá bán.
* category\_id: Khóa ngoại liên kết với danh mục.
* tax\_id: Khóa ngoại liên kết với thuế.
* thumbnail: Hình ảnh đại diện.
* is\_active: Trạng thái kích hoạt.

**Mối quan hệ:** Liên kết với categories (danh mục), taxes (thuế), product\_images (hình ảnh), product\_recipes (công thức), recipes (chi tiết công thức), và order\_details (chi tiết đơn hàng).

**Vai trò:** Quản lý danh mục sản phẩm, giá cả, thuế và công thức sản xuất. Hệ thống tự động tính toán giá sau thuế và cập nhật tồn kho dựa trên nguyên liệu sử dụng.

**Chi tiết bổ sung:** Sử dụng product\_recipes để liên kết với ingredients, hỗ trợ tính toán chi phí nguyên liệu và quản lý tỷ lệ hao hụt (waste percentage) trong công thức.

## **Ingredients (Nguyên Liệu)**

**Mô tả:** Đối tượng Ingredients đại diện cho nguyên liệu thô dùng để sản xuất sản phẩm hoặc bán trực tiếp, hỗ trợ quản lý cung ứng.

**Thuộc tính chính:**

* id: Khóa chính.
* name: Tên nguyên liệu.
* unit: Đơn vị đo lường.
* category\_id: Khóa ngoại liên kết với danh mục nguyên liệu.
* tax\_id: Khóa ngoại liên kết với thuế.
* is\_active: Trạng thái kích hoạt.

**Mối quan hệ:** Liên kết với ingredient\_categories (danh mục), taxes, recipe\_ingredients (trong công thức), supplier\_ingredient\_prices (giá từ nhà cung cấp), và branch\_ingredient\_inventory (tồn kho chi nhánh).

**Vai trò:** Quản lý từ mua hàng đến tồn kho, tính toán chi phí sản xuất và theo dõi hạn sử dụng hoặc lô hàng (batch).

**Chi tiết bổ sung:** Sử dụng inventory\_movements để ghi nhận di chuyển kho, ngưỡng tồn kho (thresholds) để cảnh báo thiếu hàng, và liên kết với purchase orders để tối ưu hóa mua sắm.

## **Suppliers (Nhà Cung Cấp)**

**Mô tả:** Đối tượng Suppliers quản lý các nhà cung cấp nguyên liệu và dịch vụ, nhằm tối ưu hóa quan hệ cung ứng.

**Thuộc tính chính:**

* id: Khóa chính.
* name: Tên nhà cung cấp.
* phone: Số điện thoại.
* email: Địa chỉ email.
* address: Địa chỉ.
* note: Ghi chú.

**Mối quan hệ:** Liên kết với ingredient\_purchase\_orders (đơn mua), supplier\_ingredient\_prices (giá), supplier\_performance (đánh giá), và purchase\_returns (trả hàng).

**Vai trò:** Theo dõi hiệu suất nhà cung cấp (tỷ lệ giao hàng đúng hạn, điểm chất lượng), xử lý trả hàng và đánh giá dựa trên đơn hàng.

**Chi tiết bổ sung:** Sử dụng supplier\_performance với các chỉ số như total\_orders, return\_value và overall\_rating để hỗ trợ quyết định chọn nhà cung cấp.

## **Customers (Khách Hàng)**

**Mô tả:** Đối tượng Customers lưu trữ thông tin khách hàng mua hàng, hỗ trợ cá nhân hóa dịch vụ.

**Thuộc tính chính:**

* id: Khóa chính (đồng nhất với user\_id).
* fullname: Họ và tên.
* phone\_number: Số điện thoại (đã mã hóa).
* email: Địa chỉ email.
* address: Địa chỉ (đã mã hóa).

**Mối quan hệ:** Liên kết với users (tài khoản) và orders (đơn hàng).

**Vai trò:** Quản lý thông tin cho đơn hàng, giao hàng và thanh toán, đảm bảo bảo mật dữ liệu cá nhân.

**Chi tiết bổ sung:** Khách hàng có thể liên kết với social\_accounts nếu đăng nhập qua Google hoặc Facebook.

## **Orders (Đơn Hàng)**

**Mô tả:** Đối tượng Orders quản lý đơn hàng bán lẻ từ khách hàng, theo dõi toàn bộ vòng đời đơn hàng.

**Thuộc tính chính:**

* id: Khóa chính.
* order\_uuid: UUID duy nhất.
* order\_code: Mã đơn hàng.
* customer\_id: Khóa ngoại liên kết với khách hàng.
* branch\_id: Khóa ngoại liên kết với chi nhánh.
* total\_money: Tổng tiền.
* status\_id: Trạng thái đơn hàng.
* notes: Ghi chú.

**Mối quan hệ:** Liên kết với order\_details (chi tiết sản phẩm), order\_payments (thanh toán), order\_shipments (vận chuyển), và order\_delivery\_tracking (theo dõi giao hàng).

**Vai trò:** Quản lý từ đặt hàng đến giao hàng, tính tổng tiền bao gồm thuế, và cập nhật trạng thái (pending, shipped, delivered).

**Chi tiết bổ sung:** Hỗ trợ theo dõi qua nhà cung cấp vận chuyển với order\_uuid và order\_code duy nhất.

## **Ingredient Purchase Orders (Đơn Mua Nguyên Liệu)**

**Mô tả:** Đối tượng Ingredient Purchase Orders xử lý đơn đặt hàng nguyên liệu từ nhà cung cấp.

**Thuộc tính chính:**

* id: Khóa chính.
* purchase\_order\_code: Mã đơn mua.
* supplier\_id: Khóa ngoại liên kết với nhà cung cấp.
* branch\_id: Khóa ngoại liên kết với chi nhánh.
* order\_date: Ngày đặt hàng.
* status\_id: Trạng thái.
* total\_amount\_before\_tax: Tổng tiền trước thuế.
* final\_amount: Tổng tiền cuối cùng.

**Mối quan hệ:** Liên kết với ingredient\_purchase\_order\_details (chi tiết), purchase\_invoices (hóa đơn), goods\_received\_notes (nhận hàng), và suppliers.

**Vai trò:** Quản lý quy trình mua hàng từ đặt hàng đến thanh toán, tính toán thuế và chiết khấu.

**Chi tiết bổ sung:** Workflow đầy đủ từ order đến invoice, GRN (goods received note), và hỗ trợ trả hàng nếu có vấn đề chất lượng.

## **Inventory (Tồn Kho)**

**Mô tả:** Đối tượng Inventory không phải là một bảng riêng mà là nhóm các bảng liên quan, quản lý tồn kho nguyên liệu tại chi nhánh và kho chính.

**Bảng chính và thuộc tính:**

* branch\_ingredient\_inventory: Tồn kho chi nhánh (quantity, safety\_stock).
* ingredient\_warehouse: Kho chính.
* inventory\_movements: Di chuyển kho (movement\_type, reference).
* inventory\_thresholds: Ngưỡng tồn kho.

**Mối quan hệ:** Liên kết với ingredients, branches, orders, purchase\_orders, và ingredient\_transfers (chuyển kho).

**Vai trò:** Theo dõi tồn kho thời gian thực, cảnh báo thiếu hàng, và ghi nhận các chuyển động (in/out/adjust). Sử dụng view v\_inventory\_status để tổng hợp dữ liệu.

**Chi tiết bổ sung:** Hỗ trợ reserved\_quantity cho đơn hàng chưa hoàn thành, expiry\_date và batch\_number để quản lý chất lượng nguyên liệu.

***Lưu ý:*** *do schema của database thật sự của bọn em thiết kế khác là to nên bọn em không đủ bỏ hết vào trang word do đó bọn em sẽ để schema database kèm chung với file báo cáo còn ở đây chỉ dùng ảnh đã đơn giản hóa.*

***A diagram of a computer

AI-generated content may be incorrect.***

Hình 1.Sơ đồ ERD rút gọn.

## **Thiết kế triển khai code**

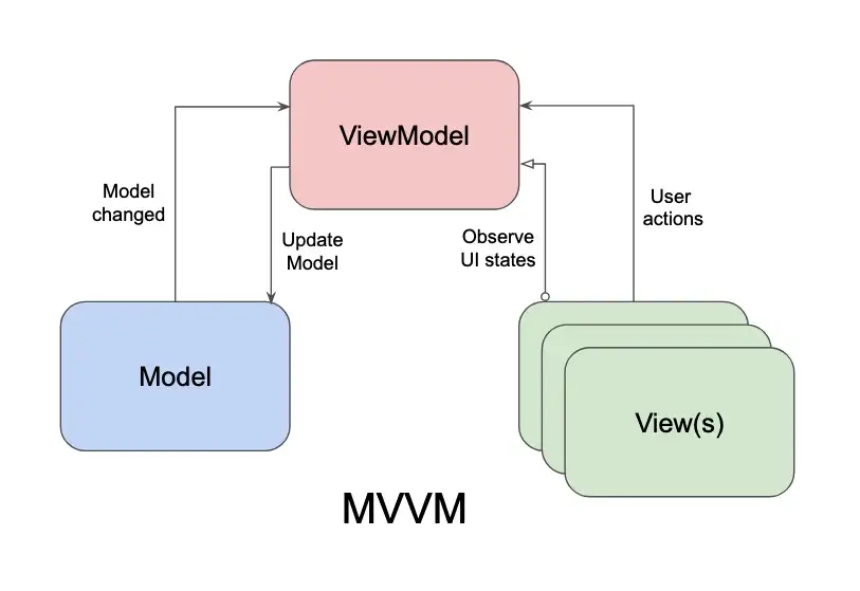
### **Tổng quan về mô hình triển khai**

#### Giới thiệu về mô hình MVVM và lý do lựa chọn

Mô hình MVVM (Model-View-ViewModel) là một kiến trúc phần mềm được phát triển để tách biệt rõ ràng các thành phần trong ứng dụng, đặc biệt phù hợp với các framework hỗ trợ data binding như winform. Cụ thể, MVVM bao gồm ba thành phần chính:

* **Model**: Đại diện cho dữ liệu và logic nghiệp vụ cốt lõi của ứng dụng, chẳng hạn như các entity dữ liệu (ví dụ: lớp SinhVien trong đồ án, chứa thông tin như mã sinh viên, họ tên, ngày sinh). Model không biết gì về giao diện người dùng, chỉ tập trung vào việc lưu trữ và xử lý dữ liệu thuần túy.
* **View**: Là lớp giao diện người dùng (UI), chịu trách nhiệm hiển thị dữ liệu và thu nhận tương tác từ người dùng. View không chứa logic xử lý, mà chỉ sử dụng data binding để liên kết với ViewModel, giúp UI trở nên "ngu ngốc" (dumb) và dễ thay đổi.
* **ViewModel**: Lớp trung gian, đóng vai trò cầu nối giữa Model và View. ViewModel xử lý các lệnh (commands) từ người dùng, chuyển đổi dữ liệu từ Model sang định dạng phù hợp để hiển thị (ví dụ: format ngày tháng), và cập nhật Model dựa trên thay đổi từ View. ViewModel thường triển khai INotifyPropertyChanged để hỗ trợ binding hai chiều.

Lý do chúng em chọn MVVM cho đồ án quản lý sinh viên xuất phát từ các đặc thù thực tế của hệ thống: Đồ án yêu cầu một giao diện động, nơi dữ liệu thay đổi thời gian thực, chẳng hạn như hiển thị danh sách sản phẩm được lọc theo loại sản phẩm hoặc tình trạng sản phẩm mà không cần reload toàn bộ trang. MVVM tận dụng data binding để đồng bộ hóa tự động, giảm thiểu code-behind trong View (thường chỉ dưới 10% code UI), từ đó tăng tốc độ phát triển và giảm lỗi. Hơn nữa, MVVM hỗ trợ kiểm thử dễ dàng hơn so với các mô hình khác như MVC, vì ViewModel có thể được unit test độc lập mà không cần mock UI. Trong bối cảnh đồ án sử dụng winform cho desktop và ASP.NET Core cho web, MVVM giúp thống nhất kiến trúc giữa các nền tảng, đảm bảo tính khả mở rộng khi thêm tính năng mới như báo cáo thống kê hoặc tích hợp API bên thứ ba. Nếu không sử dụng MVVM, hệ thống có nguy cơ rơi vào tình trạng "spaghetti code" ở UI, nơi logic nghiệp vụ lẫn lộn với hiển thị, dẫn đến khó bảo trì – một vấn đề phổ biến trong các dự án học thuật quy mô nhỏ như đồ án này.



Hình 2. Luồng đi dữ liệu của mô hình MVVM

#### Phân tầng kiến trúc với DTO, BLL, DAL và lý do tích hợp vào mvvm

Để bổ trợ cho MVVM, đồ án áp dụng kiến trúc phân tầng (Layered Architecture), bao gồm DTO (Data Transfer Object), BLL (Business Logic Layer) và DAL (Data Access Layer). Đây là một cách tiếp cận cổ điển trong phát triển phần mềm doanh nghiệp, nơi mỗi layer có trách nhiệm riêng biệt, tuân thủ nguyên tắc Separation of Concerns (SoC).

* **DTO (Data Transfer Object)**: Là các đối tượng đơn giản dùng để truyền dữ liệu giữa các layer hoặc qua mạng, không chứa logic nghiệp vụ. DTO thường là phiên bản "nhẹ" của entity, loại bỏ các trường nhạy cảm hoặc không cần thiết để tối ưu hóa serialization.

Lý do sử dụng DTO trong đồ án: Hệ thống cần trao đổi dữ liệu giữa UI (ViewModel) và backend (BLL/DAL), đặc biệt khi triển khai API RESTful sau này. Sử dụng DTO tránh lộ thông tin nội bộ (như hash mật khẩu) và giảm kích thước dữ liệu, cải thiện hiệu suất – quan trọng với dữ liệu lớn như danh sách hàng nghìn sinh viên. Không dùng DTO có thể dẫn đến "over-fetching" dữ liệu, làm chậm ứng dụng.

* **BLL (Business Logic Layer)**: Chứa toàn bộ quy tắc nghiệp vụ, xác thực và điều phối giữa các thành phần. BLL thường được triển khai qua các Services, nơi mỗi service quản lý một miền cụ thể.

Lý do áp dụng BLL: Đồ án có logic phức tạp như tính điểm trung bình tự động hoặc kiểm tra điều kiện tốt nghiệp, cần tách khỏi UI để dễ kiểm thử và tái sử dụng. BLL đảm bảo tính nhất quán nghiệp vụ, tránh duplicate code nếu không có layer này.

* **DAL (Data Access Layer)**: Xử lý tất cả tương tác với nguồn dữ liệu (CSDL như SQL Server), trừu tượng hóa qua Repository để che giấu chi tiết triển khai.

Lý do chọn DAL: Đồ án có thể thay đổi CSDL (từ SQL sang NoSQL), DAL giúp áp dụng Dependency Inversion, giảm coupling. Nó cũng hỗ trợ caching hoặc logging mà không ảnh hưởng đến BLL.

#### Các mẫu hổ trợ SOLID

**Services**: Là các lớp trong BLL triển khai logic cụ thể, thường injectable. Phân chia trách nhiệm nhỏ, dễ mở rộng (ví dụ: thêm EmailService cho thông báo). **Repository**: Mô hình trừu tượng hóa truy vấn dữ liệu, generic cho nhiều entity. Áp dụng Repository Pattern để tuân thủ DIP, dễ mock trong test. Trong đồ án, nó giúp query linh hoạt mà không hard-code SQL.

**UnitOfWork**: Quản lý transaction trên nhiều Repository, đảm bảo atomicity. Các hoạt động như cập nhật xóa các thực thể liên thông như Role và Permissio cần commit cùng lúc; UnitOfWork tránh mất đi sự nhất quán.

**Specification**: Pattern định nghĩa tiêu chí truy vấn tái sử dụng, hỗ trợ dynamic query. Đồ án cần filter phức tạp (ví dụ: như cần nhân viên A ở chi nhánh tên B); Specification cho phép soạn ra những điều kiện mà không làm Repository phức tạp.

**Dependency Injection (DI)**: Cơ chế cung cấp dependencies qua constructor, sử dụng container như .NET DI. Thúc đẩy tháo rời liên kết code, dễ test (inject mock), và quản lý scope.

Như vậy luồng đi của dữ liệu trong ứng dụng sẽ có dạng như hình dưới đây:

A diagram with colorful labels

AI-generated content may be incorrect.

Hình 1. Luồng đi dữ liệu từ khi User tương tác tới khi nhận được event

### **Tổ chức mã nguồn**

Việc tổ chức mã nguồn đóng vai trò quan trọng nhằm đảm bảo tính modular, khả bảo trì và khả mở rộng của hệ thống. Đồ án áp dụng mô hình phân tầng (Layered Architecture) với mỗi layer được triển khai dưới dạng một dự án (project) riêng biệt trong solution của Visual Studio, chẳng hạn như các project: DAL (chứa entities, Repository và UnitOfWork), Business (DTOs , BLL với Services), Common (chứa các constant và shared helper method), Winform (UI và Services viewmodel cho ứng dụng trên nền tảng windomw), và một project StockWorker riêng cho báo cáo tài chính cũng như là cảnh báo sắp hết hàng. Cách tiếp cận này không chỉ tuân thủ các nguyên tắc thiết kế phần mềm như Single Responsibility Principle (SRP) và Dependency Inversion Principle (DIP) mà còn phù hợp với đặc thù của đồ án – một hệ thống quản lý phức tạp liên quan đến dữ liệu thời gian thực như doanh thu hàng ngày, quản lý kho nguyên liệu, và báo cáo tài chính định kỳ. Dưới đây là phân tích chi tiết về lý do lựa chọn tổ chức mã nguồn theo từng project riêng biệt, kèm theo ví dụ minh họa cụ thể từ đồ án.

Lý do chính là tách biệt trách nhiệm (SoC), giảm phụ thuộc giữa các thành phần, tăng khả năng bảo trì và triển khai độc lập. Trong hệ thống F&B với dữ liệu thời gian thực, tổ chức monolithic có thể dẫn đến code khó gỡ rối và build chậm bên cạnh đó khó có thể migrate sang các ứng dụng khác như web. Các project riêng hỗ trợ referencing qua interfaces, cải thiện build hiệu suất, kiểm thử độc lập và CI/CD (ví dụ: deploy API mà không ảnh hưởng Worker). Ngoài ra, nó thúc đẩy tái sử dụng code, dễ mở rộng cho multi-tenant (mỗi cửa hàng riêng).

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Hình 3. Cấu trúc của solution.

Cấu trúc solution:

* Dashboard.Core: thư viện chung với thực thể, DTOs, interfaces (e.g., IRepository).
* Dashboard.BusinessLogic: Services (e.g., OrderService), inject từ Core/CommonAccess.
* Dashboard.CommonAccess: Repository, UnitOfWork, DbContext (EF Core cho SQL Server).
* Dashboard.WinForm: Windows Forms UI, reference BusinessLogic.
* Dashboard.StockWorker: Background service cho báo cáo kho, reference Core/CommonAccess.

### **Công nghệ sử dụng**

Trong đề tài này, ứng dụng được phát triển tập trung vào việc quản lý nội bộ phía cửa hàng cafe F&B, bao gồm các hoạt động như quản lý sản phẩm, tồn kho, nhân viên và doanh thu, mà không trực tiếp tương tác với khách hàng (phần tương tác khách hàng sẽ được xử lý bởi hệ thống web ecommerce riêng biệt). Các công nghệ được lựa chọn nhằm đảm bảo tính ổn định, hiệu suất cao và dễ triển khai, phù hợp với môi trường desktop tại cửa hàng đồng thời giúp tăng khả năng mở rộng và chuyển đổi sang nền tảng Web nếu cần. Dưới đây là tổng quan chi tiết về các công nghệ chính:

* Ngôn ngữ lập trình và nền tảng: Sử dụng C# trên .NET Core (phiên bản 8.0), một nền tảng phát triển mạnh mẽ của Microsoft hỗ trợ lập trình hướng đối tượng (OOP) và xử lý bất đồng bộ (async/await), hỗ trợ tích hợp dễ dàng với cơ sở dữ liệu và các dịch vụ chạy nền (workers). Nó cung cấp các thư viện chuẩn như MailKit cho gửi email định kỳ và System.Threading cho quản lý worker.
* Giao diện người dùng (UI): Ứng dụng sử dụng thuần WinForms – một công nghệ giao diện desktop cổ điển của .NET, cho phép tạo các form, control và event handler đơn giản, hiệu quả, giúp giảm độ phức tạp và đảm bảo ứng dụng chạy mượt mà trên các máy tính cấu hình trung bình tại cửa hàng cafe F&B. Các thành phần UI bao gồm DataGridView cho hiển thị danh sách, TextBox/ComboBox cho nhập liệu, và Chart control cho báo cáo thống kê.
* Cơ sở dữ liệu: Sử dụng SQL Server (phiên bản 2022) làm cơ sở dữ liệu chính, hỗ trợ lưu trữ dữ liệu lớn với tính năng như stored procedures, triggers và indexing để tối ưu hóa truy vấn. Để dễ triển khai, quản lý và cũng như hỗ trợ quá trình phát triển của nhóm, SQL Server được host trên Docker – một nền tảng container hóa cho phép chạy cơ sở dữ liệu trong môi trường ảo hóa, đảm bảo tính di động và dễ scale. Docker cũng được sử dụng để host các worker chạy nền, giúp tách biệt môi trường phát triển và sản xuất mà không phụ thuộc vào máy chủ vật lý.
* Công cụ hỗ trợ và thư viện:
  + Entity Framework (EF) Core làm ORM (Object-Relational Mapping) để kết nối giữa code C# và SQL Server, hỗ trợ migration dữ liệu và truy vấn LINQ hiệu quả.
  + Các thư viện chuẩn của .NET cho xử lý bất đồng bộ (System.Threading.Tasks) và gửi email (MailKit).
  + Docker Compose để orchestrate các container cho SQL Server và workers, đảm bảo tính nhất quán trong triển khai.
  + AutoMapper (13.0.1): Mapping objects giữa DTOs/Entities. Nổi bật ở simplicity, giảm code lặp. Trong đồ án, map InventoryEntity sang ReportDto cho Worker: mapper.Map<WeeklyStockReportDto>(entity).

Bên cạnh đó do tổ chức mã nguồn phân lớp theo các dự án (.csprj) do đó việc đồng bộ hóa các package version để tránh gặp xung đột cũng cần được cân nhắc, do đó ở đây chúng em sử dụng CPM thông qua một Directory.Build.props duy nhất để có thể quy định các version của toàn bộ các dự án sử dụng package phải tuân theo. Các công nghệ này được tích hợp tăng hiệu quả công việc cải thiện quá trình lập trình mã nguồn của nhóm. A computer screen shot of a program

AI-generated content may be incorrect.

Hình 4. Quản lý phiên bản gói tập trung

# **CHƯƠNG III: THIẾT KẾ GIAO DIỆN HỆ THỐNG**

## **Đăng nhập**

**Mục đích:** Cho phép người dùng đăng nhập vào hệ thống với tài khoản và mật khẩu. Mô tả giao diện: Gồm các trường nhập username, password và nút Đăng nhập, Thoát.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 5. Giao diện đăng nhập.

**Tính năng chính:**

* Kiểm tra dữ liệu nhập hợp lệ.
* Hiển thị thông báo lỗi nếu sai thông tin.
* Cho phép chuyển tới màn hình trang chủ khi đăng nhập thành công.
* Xử lý session và token để thực hiện giới hạn quyền trong ứng dụng.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 6. Hiển thị thông báo lỗi thông qua toast message khi người dùng lỗi hoặc server lỗi

A long thin brown stick

AI-generated content may be incorrect.

Hình 7. Cho phép hiển thị mật khẩu để dễ dàng nhập hơn hoặc tắt đi để đảm bảo an toàn

A screenshot of a login screen

AI-generated content may be incorrect.

Hình 8. Chỉ hiệu đang tải giúp người dùng nhận biết tình trạng ứng dụng đồng thời ngăn người dùng tạo thêm sự kiện mới gây xun đột

## **Thanh điều hướng bên hông**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 9. Thanh điều hướng.

Cho phép thực hiện chuyển sang các form khác một cách dễ dàng. Ngoài ra khi ấn vào các Icon trong hình có thể thực hiện mở rộng hoặc thu nhỏ phần sidebar như hình bên dưới

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Hình 10. Các thành pần có thể thực hiện đóng mở

## **Trang chủ**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 11. Giao diện trang chủ.

**Mục đích:** Trang chủ tổng quan cung cấp thông tin nhanh về số đơn hàng, doanh thu, lợi nhuận, tình trạng hàng hóa và các số liệu quan trọng khác giúp người quản lý có cái nhìn tổng thể về hoạt động kinh doanh trong các khoảng thời gian khác nhau.

**Tính năng chính:**

* Cho phép người dùng chọn khoảng thời gian để xem số liệu tương ứng.
* Hiển thị các chỉ số quan trọng một cách trực quan, dễ hiểu.
* Cung cấp biểu đồ so sánh doanh thu, chi phí và lợi nhuận để phân tích nhanh hiệu quả kinh doanh.
* Hiển thị nhanh số lượng khách hàng, nhà cung cấp và sản phẩm hiện có.
* Cảnh báo hàng hóa sắp hết tồn kho để người quản lý kịp thời nhập thêm.

## **Giao diện Danh sách chung**

**Mô tả:** Giao diện danh sách chung có chức năng hiển thị các bản ghi của các thực thể mà hệ thống có nhiệm vụ quản lý. Trong màn hình này, có các thành phần hỗ trợ người dùng có thể dễ dàng tìm kiếm lọc thông tin một cách dễ dàng.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 12. Mô tả giao diện hiển thị danh sách.

**Chú thích:** ngoài những thành phần đã hiển thị rõ ràng như

* Vùng màu vàng: Cho phép thực hiện tìm kiếm Bound data, filter dữ liệu theo trường dữ liệu tùy ý một cách nhanh chóng.
* Vùng màu đỏ (Header): cho phép thực hiện sắp xếp bảng ghi dựa theo cột tùy chọn.
* Vùng màu hồng (các combo box): cho phép thực hiện lọc dữ liệu theo thuộc tính hoặc sắp xếp theo một số trường nhất định.
* Vùng màu xanh: có thể nhấp đúp chuột để mở xem chi tiết thông tin hoặc có thể bấm chuột phải để thực hiện tải lại dữ liệu nếu nghi ngờ có sai sót.
* Vùng màu cam: hiển thị số lượng bảng ghi trong tập dữ liệu thực thể đó, cho phép lựa chọn số lượng bản ghi hiển thị 1 lần và chuyển trang trước sau.

## **Nhân sự**

**Mục đích:** Quản lý thông tin cơ bản của nhân viên trong công ty, cho phép thêm mới, chỉnh sửa và theo dõi các thông tin cá nhân, vị trí làm việc và trạng thái của nhân viên.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 13. Danh sách nhân sự.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 14. Màn hình thêm/chỉnh sửa nhân viên.

**Mô tả:** Ở trang đầu tiên ta có thể xem chi tiết thông tin nhân viên, có thể thực hiện chỉnh sửa cho tới khi ấn lưu để xác nhận ghi thông tin chỉnh sửa, hoặc hủy để loại bỏ thay đổi. Ở trang thứ hai (như hình bên dưới) cho phép xem thông tin liệu nhân viên đó có tài khoản liên kết nào để truy cập vào hệ thống hay không (vì có những nhân viên chỉ làm việc ở bộ phận ngoài không liên quan đến hệ có những nhân viên có tham gia hệ thống).

A screenshot of a computer

AI-generated content may be incorrect.

Hình 15. Thông tin về tên tài khoản và vai trò của người dùng trong hệ thống.

* Thanh menu bên trái có mục **Nhân sự**, khi chọn sẽ hiển thị danh sách nhân viên hiện có.
* Giao diện chính gồm các trường thông tin như:
  + Mã chi nhánh, Họ tên, Chức vụ, Số điện thoại, Trạng thái hoạt động.
* Phía trên có các bộ lọc và tìm kiếm theo ID, trạng thái, vai trò, tên/ID nhân viên.
* Nút chức năng: **Chi tiết** (xem thông tin chi tiết), **Thêm** (thêm nhân viên mới).
* Khi thêm nhân viên mới, giao diện mở ra cửa sổ với nhiều tab:
  + **Thông tin cơ bản:** ID nhân viên, Họ tên đầy đủ, Số điện thoại, Email, Chi nhánh, Chức vụ, Trạng thái (Active/Inactive), Ngày tuyển dụng, Ngày nghỉ việc, Ngày tạo, Ngày cập nhật.
  + **Thông tin tài khoảnTính năng chính:**
* Thêm mới nhân viên với đầy đủ thông tin.
* Cập nhật thông tin nhân viên hiện có.
* Quản lý trạng thái nhân viên (đang làm việc, nghỉ việc).
* Theo dõi ngày tuyển dụng, nghỉ việc và các dữ liệu liên quan khác.

## **Tài khoản**

**Mô tả:** Quản lý tài khoản đăng nhập hệ thống, phân quyền và gán vai trò cho từng người dùng (lưu ý tài khoản người dùng nội bộ và tài khoản của khách hàng đã tách riêng ra còn dữ liệu trong ảnh là dữ liệu chỉ dùng để thử nghiệm).

A screenshot of a computer

AI-generated content may be incorrect.

Hình 16. Danh sách tài khoản.

**Thêm/sửa người dùng**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 17. Giao diện thêm / sửa người dùng.

**Gán vai trò**

A computer screen shot of a blue screen

AI-generated content may be incorrect.

Hình 18. Giao diện cho phép thực hiện gán vai trò cho người dùng.

**Mô tả giao diện:**

* Giao diện gồm ba tab chính:
  + **Danh sách người dùng:** Hiển thị danh sách tài khoản người dùng với các thông tin ID User, Tên đăng nhập, Tên nhân viên, Vai trò, Trạng thái, Ngày tạo.
  + **Thêm/Sửa người dùng:** Cho phép tạo mới hoặc chỉnh sửa thông tin tài khoản.
  + **Gán vai trò:** Phân quyền vai trò cho người dùng trong hệ thống.
* Có các bộ lọc để tìm kiếm theo vai trò, trạng thái, ID, tên đăng nhập.
* Các nút thao tác:
  + **Chi tiết:** Xem chi tiết thông tin người dùng.
  + **Thêm User:** Thêm tài khoản mới.
  + **Sửa:** Chỉnh sửa tài khoản hiện tại.
* Khu vực danh sách hiển thị trạng thái hoạt động của tài khoản và ngày tạo.

**Tính năng chính:**

* Tạo tài khoản người dùng mới.
* Sửa thông tin tài khoản hiện tại.
* Gán vai trò, phân quyền sử dụng hệ thống cho người dùng.
* Tìm kiếm và lọc tài khoản theo nhiều tiêu chí.

## **Vai trò và quyền**

Mô tả: Ở trang chính (tab danh sách) hiển thị danh sách Role (phía trên) và danh sách Permission (phía dưới). Người dùng có thể chọn một role để xem hoặc chuyển sang tab gán quyền; chỉnh sửa hoặc thêm mới chỉ được xác nhận khi nhấn Lưu, hoặc hủy để loại bỏ thay đổi. Có các tab riêng để Thêm/Sửa Role, Thêm/Sửa Permission và Gán quyền cho Role.

A computer screen shot of a program

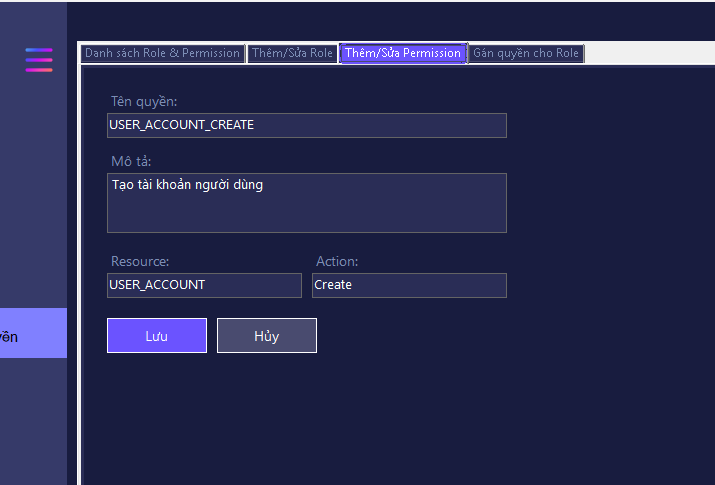
AI-generated content may be incorrect.

Hình 19. Danh sách vai trò và quyền.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 20. Trang thêm / sửa vai trò



Hình 21. Trang thêm sửa quyền hạn

A screenshot of a computer

AI-generated content may be incorrect.

Hình 22. Trang gán quyền hạn cho vai trò

* Thanh menu bên trái có mục Role/Permission, khi chọn sẽ mở giao diện quản lý này.
* Giao diện chính gồm các trường/ cột hiển thị: ID, Tên Role/Permission, Mô tả, Resource, Action, Số quyền (Role), Ngày tạo.
* Phía trên có các bộ lọc/tìm kiếm và sắp xếp (theo ID, tên, ngày tạo, …) và phân trang do presenter quản lý.
* Nút / menu chức năng: Chi tiết (mở tab sửa), Thêm (mở tab thêm mới), Xóa (xóa role/permission), Gán quyền (mở tab gán quyền).

Khi Thêm / Sửa:

* Mở tab Thêm/Sửa Role: trường Tên, Mô tả (và có thể gán quyền khi cần).
* Mở tab Thêm/Sửa Permission: trường Tên, Mô tả, Resource, Action.
* Nút Lưu: gọi service lưu tạo hoặc cập nhật; Nút Hủy: quay lại danh sách, bỏ thay đổi.

Tab Gán quyền cho Role:

* Hiển thị tên Role đã chọn; hai bảng song song: Quyền có sẵn (Available) và Quyền đã gán (Assigned).
* Giữa hai bảng có nút → để gán nhiều quyền, nút ← để gỡ quyền.
* Sau gán/gỡ, presenter cập nhật số quyền của role và làm mới danh sách.

Tính năng chính (người dùng có thể làm):

* Xem danh sách Role và Permission.
* Tìm kiếm, sắp xếp và phân trang danh sách.
* Thêm mới / sửa / xóa Role và Permission.
* Gán hoặc gỡ nhiều Permission cho 1 Role (chọn nhiều hàng, nhấn → / ←).
* Mở nhanh form sửa bằng double‑click hoặc menu ngữ cảnh.
* Xác nhận hành động quan trọng qua hộp thoại (xóa, gán/gỡ).

## **Hàng hóa**

**Mục đích:** Quản lý thông tin nguyên liệu, hàng hóa trong kho của công ty, cho phép thêm mới, chỉnh sửa, tìm kiếm và theo dõi trạng thái hoạt động của từng nguyên liệu.

**Danh sách nguyên liệu:**

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 23. Danh sách hàng hóa.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 24. Thêm nguyên liệu

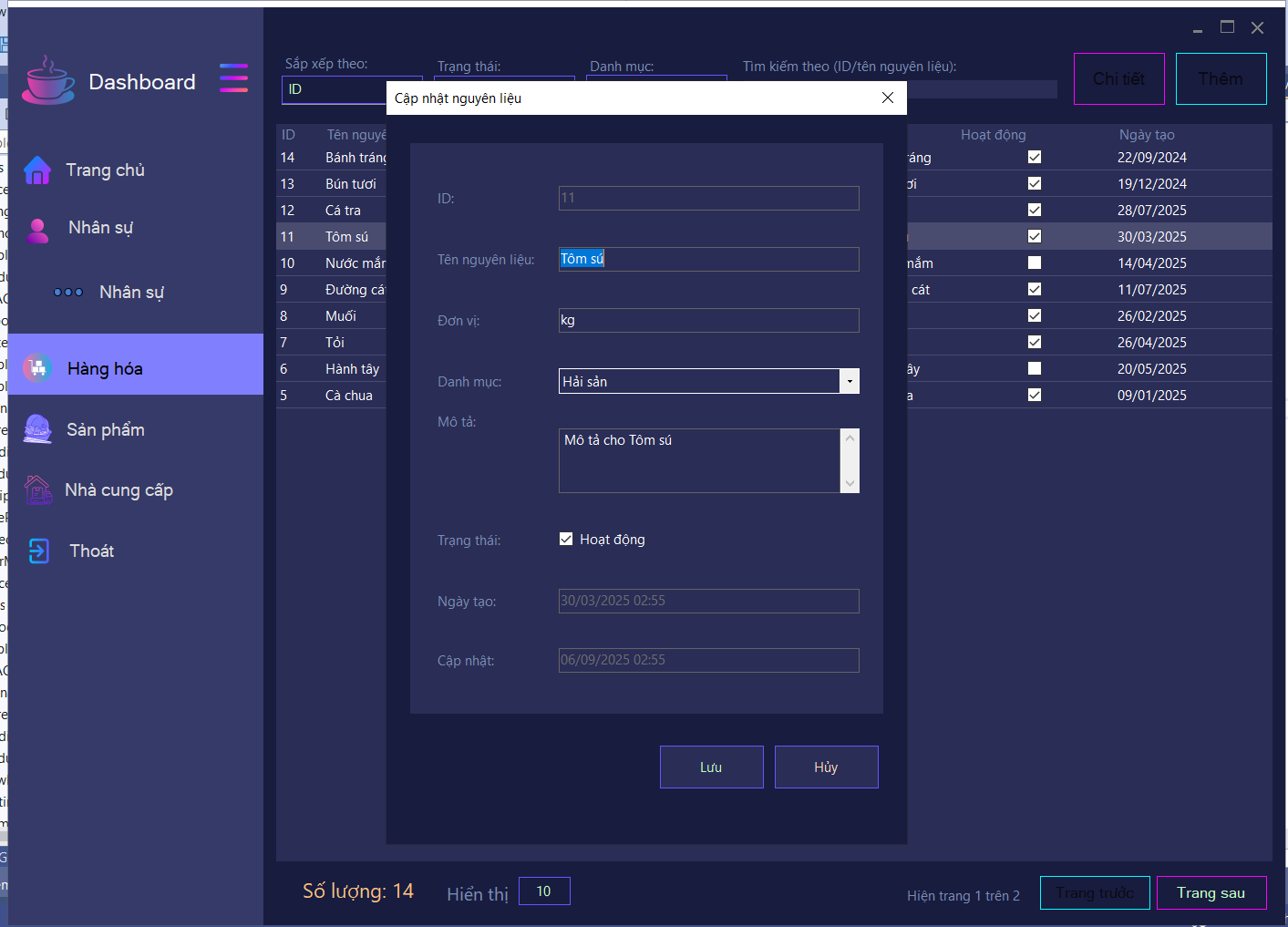
Hiển thị bảng danh sách các nguyên liệu với các thông tin bao gồm:  
ID, Tên nguyên liệu, Danh mục, Đơn vị, Mô tả, Trạng thái hoạt động, Ngày tạo.

**Các bộ lọc tìm kiếm:**

* Cho phép lọc nguyên liệu theo các tiêu chí:
* Sắp xếp theo ID
* Trạng thái (Hoạt động hoặc không hoạt động)
* Danh mục nguyên liệu
* Tìm kiếm theo ID hoặc tên nguyên liệu

**Giao diện form thêm nguyên liệu bao gồm các trường:**

* Tên nguyên liệu (bắt buộc nhập)
* Đơn vị (bắt buộc nhập)
* Danh mục (Chọn từ danh sách có sẵn, ví dụ: Thịt, Hải sản, Rau củ, Gia vị, Đồ khô)
* Mô tả (có thể nhập thêm thông tin chi tiết)
* Trạng thái (checkbox chọn Hoạt động hoặc không)

**Chỉnh sửa nguyên liệu**

Hình 25. Chi tiết nguyên liệu và cập nhật thông tin

**Giao diện form chỉnh sửa tương tự như form thêm, hiển thị thêm các thông tin:**

* ID (không sửa được)
* Ngày tạo (không sửa được)
* Ngày cập nhật (tự động cập nhật khi chỉnh sửa)
* Người dùng có thể thay đổi tên, đơn vị, danh mục, mô tả và trạng thái hoạt động.

## **Nhà cung cấp**

**Mục đích:** Nhằm giúp người dùng quản lý toàn diện thông tin sản phẩm bao gồm thông tin cơ bản cơ bản về nhà cung cấp. Giao diện cho phép xem, thêm, sửa, xóa dữ liệu để đảm bảo cập nhật các thông tin.

Danh sách nhà cung cấp

A screenshot of a computer

AI-generated content may be incorrect.

Hình 26. Danh sách nhà cung cấp.

**Bảng hiển thị các cột:**

* ID: Mã định danh nhà cung cấp
* Tên sản phẩm: Tên nhà cung cấp
* Số điện thoại: số điện thoại liên hệ đặt hàng
* Địa chỉ: địa chỉ của nhà cung cấp
* Ghi chú: ghi chú nếu có ưu tiên hay đánh giá nội bộ về nhà cung cấp cho sau này.
* Ngày tạo phiếu và ngày cập nhật.

**Tìm kiếm, lọc, sắp xếp**

* Sắp xếp theo: Có thể chọn sắp xếp theo các trường, ví dụ ID, Tên, Số điện thoại, ...
* Trạng thái: Lọc sản phẩm theo trạng thái (All / Hoạt động / Không hoạt động).
* Loại nhà cung cấp: Lọc theo loại sản phẩm mà nhà cung cấp cung cấp.
* Tìm kiếm theo (ID/tên sản phẩm): Nhập từ khóa để tìm kiếm sản phẩm theo ID hoặc tên.

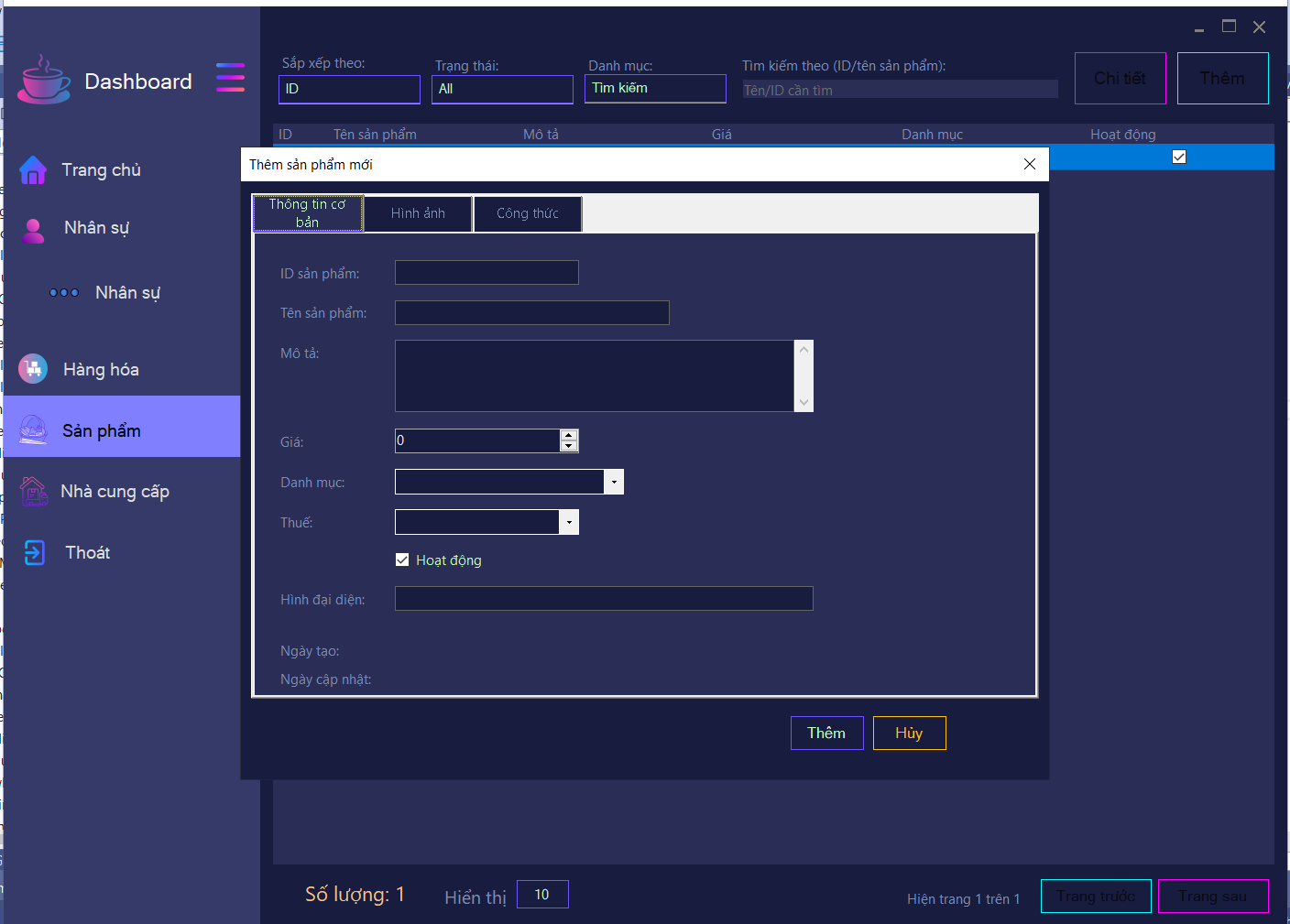
**Thêm/chỉnh sửa sản phẩm**

* **Tab Thông tin cơ bản**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 27. Chi tiết thông tin về nhà cung cấp (có thể thêm sửa hoặc xóa).



**Xem:** Hiển thị đầy đủ các thông tin sản phẩm như tên, mô tả, giá, danh mục, trạng thái hoạt động.

**Thêm:**

* Nhập dữ liệu mới cho tất cả trường thông tin.
* Nhấn Cập để lưu.

**Sửa:**

* Chỉnh sửa các trường thông tin hiện có.
* Nhấn Cập để lưu.

## **Sản phẩm**

**Mô tả:** Cho phép người dùng có thể thực hiện thêm sửa xóa và tìm kiếm thông tin giúp dễ dàng quản lý các sản phẩm và công thức trong cửa hàng.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 28. Giao diện danh sách thông tin sản phẩm.

**A screenshot of a computer

AI-generated content may be incorrect.**

**Hình 29. Giao diện chi tiết sản phẩm.**

**Trên trang danh sách sản phẩm (dgvProducts):**

* Xem danh sách sản phẩm, chọn 1 hàng để thao tác.
* Chọn 1 sản phẩm rồi mở chi tiết/ chỉnh sửa (double‑click hoặc nút/ menu ngữ cảnh nếu có).
* (Giao diện hỗ trợ chế độ chỉ đọc trên grid; các thao tác tạo/sửa/xóa sẽ mở form chi tiết).

**Mở form Chi tiết / Thêm sản phẩm (FrmProductDetails):**

* Xem/ chỉnh sửa thông tin cơ bản: Tên, Mô tả, Giá, Trạng thái hoạt động.
* Chọn danh mục và thuế từ combobox (nếu không có dữ liệu sẽ hiển thị giá trị mặc định).
* Thêm sản phẩm mới (form ở chế độ Thêm) hoặc lưu thay đổi cho sản phẩm hiện có (chế độ Sửa) bằng nút Lưu/Thêm.
* Hủy/Đóng form bằng nút Hủy hoặc Đóng mà không lưu.

**Quản lý ảnh sản phẩm:**

* Upload ảnh từ file: sao chép tạm vào temp và hiển thị preview.
* Nhập URL hoặc đường dẫn ảnh bằng tay vào trường và kiểm tra tính hợp lệ.
* Xem ảnh đại diện/preview — load từ URL hoặc file cục bộ.
* Xóa ảnh trong UI — chỉ xóa trên giao diện; ghi thực tế khi nhấn Lưu.
* Loại bỏ ảnh hiện tại để xóa vùng preview và đặt ảnh rỗng.
* Khi lưu: ảnh từ URL hoặc file sẽ được tải/ sao chép vào thư mục Resources/Uploads (nếu cần) trước khi gửi lên service.

**Xác thực và trạng thái lưu:**

* Kiểm tra URL ảnh — hiển thị trạng thái kiểm tra (Đang kiểm tra / Hợp lệ / Không hợp lệ).
* Nút Lưu chỉ hoạt động khi ảnh (nếu thay đổi) đã được kiểm tra và hợp lệ.

Trong khi thao tác mạng/ lưu, giao diện bị khóa (SetLoadingState) và con trỏ hiển thị chờ.

**Quản lý danh sách hình ảnh sản phẩm:**

* Bảng ảnh hiển thị ID, URL, Ngày tạo.
* Chọn một ảnh trong bảng để xem hoặc để làm ảnh đại diện.
* Xóa ảnh: ảnh bị gỡ khỏi danh sách local và đánh dấu id để xóa khi lưu.

**Quản lý công thức liên quan (Recipes):**

* Danh sách công thức có sẵn trong combobox; chọn và gán công thức cho sản phẩm.
* Bảng “Công thức đã gán” hiển thị các công thức hiện có; chọn để bỏ gán.
* Tạo công thức mới: mở form tạo, nếu tạo thành công sẽ thêm vào danh sách có sẵn.
* Sửa công thức đã gán: mở form sửa; cập nhật thay đổi trên bảng assigned.
* Xem chi tiết công thức ở chế độ chỉ xem.
* Khi lưu sản phẩm: các thay đổi gán/bỏ gán công thức sẽ được lưu (Assign/Unassign gọi presenter tương ứng).

**Validations & thông báo:**

* Kiểm tra trường bắt buộc: Tên sản phẩm không được rỗng; Giá phải > 0; phải chọn danh mục.
* Các thông báo lỗi/ thành công hiển thị thông báo.
* Xác nhận trước khi xóa **ảnh hoặc bỏ gán công thức.**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 30. Danh sách hình ảnh của một sản phẩm.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 31. Danh sách chi tiết càc công thức liên qua

## **Hệ thống gửi email thông báo**

* 1. Worker thông báo sắp hết hàng

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a chat

AI-generated content may be incorrect.

Hình 32. Giao diện email thông báo hết hàng.

Giao diện email cảnh báo các nguyên liệu sắp hết hàng.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 33. File csv gửi đính kèm thông báo

* 1. Worker báo cáo doanh thu định kì

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a chat

AI-generated content may be incorrect.

* A screenshot of a computer

  AI-generated content may be incorrect.

# **CHƯƠNG IV: TRIỂN KHAI MÃ NGUỒN VÀ THIẾT KẾ LOGIC**

## **Về tổng quan thiết kế mã nguồn**

Trong solution này mã nguồn được thiết kế theo mô hình 3 lớp với MVVM làm chủ đạo. Bởi vì MVVM giải quyết vấn đề tách biệt logic khỏi giao diện tốt.

**- Tách biệt UI và Logic**

* UI (View) chỉ lo hiển thị.
* ViewModel chứa state + logic trình bày.
* Model chứa dữ liệu gốc (domain, DTO, entity).  
  → Giúp code **dễ bảo trì, dễ thay đổi giao diện** mà không ảnh hưởng tới logic.

**- Data Binding & Two-way Binding**

* View và ViewModel kết nối qua **data binding**.
* Khi ViewModel thay đổi property → UI tự cập nhật.
* Khi người dùng nhập liệu trên UI → ViewModel cũng được update.  
  → Giảm kết dính mã nguồn kiểu textBox.Text = model.Name;
* Bên cạnh đó là có thể dễ dàng thực hiện xử lý data bound

**- Tái sử dụng**

* Một ViewModel có thể dùng với nhiều Form khác nhau (list, form, dialog).
* Dễ viết component UI tùy chỉnh (reusable controls).

Trong đó các tầng UI, miền logic và truy vấn dữ liệu được chia thành Libray Project như hình bên dưới đây.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.

Hình 34. Cấu trúc dự án.

Ngoài ra để đảm bảo khả năng mở rộng và tái tổ chức cho sau này, dự án còn sử dụng một số Mẫu lập trình như sau:

**Service Pattern**

* **Mục đích:** Tách **business logic** ra khỏi UI và DataAccess.
* **Công dụng:**
  + Gom nhóm các **nghiệp vụ (use case)** thành các lớp/service riêng (vd: UserService, OrderService).
  + Đảm bảo UI (Winform, API) chỉ gọi service mà không phải xử lý logic phức tạp.
  + Tăng tính **tái sử dụng** và **testable** (dễ viết unit test cho service).

**Repository Pattern**

* **Mục đích:** Tách logic truy xuất dữ liệu (EF, SQL, MongoDB, API…) ra khỏi business logic.
* **Công dụng:**
  + Đóng vai trò như **kho chứa** (repository) quản lý entity.
  + Giúp code business không phụ thuộc EF Core hoặc SQL trực tiếp.
  + Cho phép thay đổi DB dễ dàng (vd đổi EF sang Dapper mà service không cần sửa).

**Unit of Work Pattern**

* **Mục đích:** Quản lý transaction khi làm việc với nhiều repository.
* **Công dụng:**
  + Gom nhiều thao tác DB thành một transaction duy nhất.
  + Đảm bảo tính toàn vẹn dữ liệu (nếu lỗi giữa chừng → rollback).
  + Giúp kiểm soát SaveChanges ở một chỗ, thay vì gọi lẻ tẻ trong từng repository.

**Specification Pattern**

* **Mục đích: Tách biệt logic lọc/truy vấn (query criteria) khỏi repository.**
* **Công dụng:**
  + Thay vì viết LINQ query trực tiếp trong service → đưa vào Specification.
  + Tái sử dụng điều kiện lọc phức tạp (vd: "tất cả đơn hàng chưa thanh toán trong 7 ngày").
  + Làm code repository + service gọn hơn.
  + Giúp test query dễ dàng.

## **2. Tổ chức mã nguồn trong Solution**

Solution được chia thành nhiều **Library Project**, mỗi project đảm nhận một nhiệm vụ riêng:

* **Dashboard.Core**: Chứa entity, DTO, interface (IRepository, ISpecification...).
* **Dashboard.CommonAccess**: Quản lý DbContext, Repository, UnitOfWork, migration cho EF Core.
* **Dashboard.BusinessLogic**: Chứa Service xử lý nghiệp vụ (UserService, ProductService, InventoryService...).
* **Dashboard.WinForm**: Lớp giao diện người dùng, gồm các Form kết hợp với ViewModel và binding dữ liệu.
* **Dashboard.StockWorker**: Dịch vụ chạy nền, thực hiện các job định kỳ như gửi báo cáo doanh thu, cảnh báo tồn kho.

Cách chia này tuân thủ **Separation of Concerns (SoC)** và nguyên tắc **SOLID**, đảm bảo hệ thống dễ bảo trì, mở rộng, và hỗ trợ triển khai CI/CD sau này.

## **3. Luồng xử lý nghiệp vụ**

Ví dụ minh họa quy trình **tạo đơn hàng**:

1. Người dùng nhập thông tin trên Form (View).
2. ViewModel kiểm tra dữ liệu và gọi **Presenter** thực hiện gọi các thao tác nghiệp vụ xử lý yêu cầu từ người dùng.
3. Giả sử nghiệp vụ khi này là lấy một danh sách sản phẩm trình chiếu lên Form, sử dụng **UnitOfWork + Repository** để lưu đơn hàng và trừ tồn kho.
4. EF Core cập nhật dữ liệu xuống SQL Server.
5. Kết quả trả về ViewModel → UI hiển thị thông báo và refresh danh sách đơn hàng.

Cách tiếp cận này giúp tách biệt rõ trách nhiệm, tăng khả năng kiểm thử và dễ dàng mở rộng nghiệp vụ.

# **CHƯƠNG V: ĐÁNH GIÁ HỆ THỐNG**

## **1. Ưu điểm**

* **Kiến trúc rõ ràng**: Hệ thống được thiết kế theo mô hình 3 lớp kết hợp MVVM và các mẫu thiết kế (Service, Repository, Unit of Work, Specification) → dễ bảo trì, dễ mở rộng.
* **Phân quyền chi tiết**: Cơ chế RBAC (Role-Based Access Control) cho phép quản lý truy cập theo vai trò (Admin, Manager, Cashier, Employee…), hạn chế sai sót và tăng tính bảo mật.
* **Giao diện trực quan**: Sử dụng WinForms kết hợp Data Binding, giúp thao tác đơn giản, hỗ trợ lọc – tìm kiếm – phân trang.
* **Dashboard hiệu quả**: Hiển thị số liệu doanh thu, đơn hàng, tình trạng tồn kho theo thời gian thực, hỗ trợ quản lý ra quyết định nhanh.
* **Tự động hóa**: Worker chạy nền định kỳ, tự động gửi báo cáo doanh thu hàng tháng và cảnh báo tồn kho thấp qua email, giúp giảm thao tác thủ công.
* **Khả năng mở rộng**: Việc có sẵn API trong module dashboard-api cho phép dễ dàng tích hợp thêm Web/Mobile Dashboard trong tương lai.

## **2. Hạn chế**

* **Quy mô nhỏ**: Mới thử nghiệm trên dữ liệu mô phỏng, chưa triển khai trên hệ thống nhiều chi nhánh thực tế.
* **Giao diện WinForms**: Dù ổn định nhưng còn hạn chế về tính hiện đại so với Web App/ Mobile App.
* **Thiếu tích hợp thanh toán điện tử**: Hệ thống mới chỉ quản lý nội bộ, chưa hỗ trợ các cổng thanh toán như Momo, VNPay.
* **Chưa có real-time notification**: Thông báo mới dựa vào worker định kỳ, chưa hỗ trợ đẩy thông báo tức thì (push notification).
* **Báo cáo phân tích**: Dashboard mới dừng ở mức thống kê cơ bản, chưa có phân tích chuyên sâu (BI, Data Analytics).

# **CHƯƠNG VI: HƯỚNG PHÁT TRIỂN**

Trong tương lai, nhóm định hướng phát triển hệ thống theo các hướng sau:

1. **Phát triển Dashboard Web**
   * Sử dụng React/Next.js hoặc Angular kết hợp API ASP.NET Core để quản lý từ xa qua trình duyệt.
   * Tích hợp bảo mật OAuth2.0/SSO để tăng độ an toàn khi truy cập.
2. **Mở rộng kênh bán hàng (E-commerce)**
   * Xây dựng website/ứng dụng đặt hàng trực tuyến cho khách hàng.
   * Kết nối trực tiếp với hệ thống quản lý nội bộ để đồng bộ đơn hàng và tồn kho.
3. **Tích hợp thanh toán online**
   * Liên kết các ví điện tử phổ biến tại Việt Nam (Momo, ZaloPay, VNPay).
   * Hỗ trợ thanh toán qua QR Code và thẻ tín dụng.
4. **Phân tích nâng cao (Business Intelligence)**
   * Tích hợp Power BI hoặc Grafana để trực quan hóa dữ liệu.
   * Phân tích hành vi khách hàng, dự đoán doanh thu, tối ưu tồn kho.
5. **Ứng dụng Machine Learning**
   * Dự đoán nhu cầu nguyên liệu dựa trên dữ liệu bán hàng lịch sử.
   * Gợi ý upsell/cross-sell để tăng doanh thu.
6. **Mở rộng quy mô đa chi nhánh**
   * Triển khai hệ thống trên môi trường cloud (Azure/AWS).
   * Đồng bộ dữ liệu theo thời gian thực giữa nhiều chi nhánh.

# **KẾT LUẬN**

Đề tài **“Xây dựng hệ thống quản lý cửa hàng cafe F&B”** đã đạt được những mục tiêu đặt ra ban đầu:

* Xây dựng được ứng dụng quản lý sản phẩm, tồn kho, nhân viên, đơn hàng và báo cáo doanh thu.
* Triển khai thành công Dashboard API và các worker chạy nền, góp phần nâng cao tính tự động hóa trong quản lý.
* Ứng dụng kiến trúc phần mềm hiện đại (MVVM + Layered Architecture + Repository + Unit of Work), đảm bảo tính ổn định, dễ bảo trì và mở rộng.

Mặc dù còn một số hạn chế về giao diện, khả năng mở rộng và phân tích dữ liệu, nhưng đề tài đã chứng minh tính khả thi của việc áp dụng các công nghệ .NET, SQL Server và WinForms vào quản lý cửa hàng F&B.

Kết quả đạt được không chỉ giúp nhóm củng cố kiến thức lập trình và thiết kế hệ thống, mà còn mở ra hướng phát triển thành một giải pháp phần mềm hoàn chỉnh, có thể áp dụng thực tiễn trong các cửa hàng cà phê, nhà hàng và lĩnh vực F&B nói chung.

# **TÀI LIỆU THAM KHẢO**

[1] “Highlands Coffee: Brewing Success in Vietnam,” B2Bnn, 2025. [Online]. Available: https://b2bnn.com/highlands-coffee-success-vietnam

[2] P. H. Trang, “A Strategic Analysis of Highlands Coffee's Success,” *ResearchGate*, 2024. [Online]. Available: <https://www.researchgate.net/publication/HighlandsCoffee>

[3] Microsoft, “Windows Forms Overview,” *Microsoft Docs*, 2025. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/desktop/winforms

[4] Microsoft, “Entity Framework Core Documentation,” *Microsoft Docs*, 2025. [Online]. Available: https://learn.microsoft.com/en-us/ef/core

[5] Microsoft, “ASP.NET Core MVC Overview,” *Microsoft Docs*, 2025. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/mvc

[6] Microsoft, “SQL Server 2022 Technical Documentation,” *Microsoft Docs*, 2025. [Online]. Available: https://learn.microsoft.com/en-us/sql

[7] M. Fowler, “Unit of Work Pattern,” *martinfowler.com*, 2025. [Online]. Available: https://martinfowler.com/eaaCatalog/unitOfWork.html

[8] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley, 2004.

[9] DigitalOcean, “Docker Compose Tutorial,” *DigitalOcean Community*, 2025. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose

[10] Automapper, “Automapper Documentation,” *Automapper.org*, 2025. [Online]. Available: https://docs.automapper.org/en/stable

# **PHỤ LỤC HƯỚNG DẪN CHẠY VÀ TÀI KHOẢN**

Tài khoản đăng nhập

Username: taosieudeptrai

Password: taosieudeptrai

Hướng dẫn chạy

**Đối với worker**

Vào thư mục thực hiện config Dashboard.Worker/appsetting.json

theo mẫu hướng dẫn bên dưới, lưu ý đổi connection string thành connection string thật sự

{

// =======================

// KẾT NỐI DATABASE

// =======================

"ConnectionStrings": {

"WebbanhangDB": "Server=127.0.0.1,1433;Database=YourDatabaseName;User Id=YourUsername;Password=YourPassword;Encrypt=true;TrustServerCertificate=true"

// 🔹 Server=...: Địa chỉ + cổng DB (SQL Server)

// 🔹 Database=...: Tên cơ sở dữ liệu

// 🔹 User Id / Password: Tài khoản đăng nhập DB

// 🔹 Encrypt=true;TrustServerCertificate=true: Cho phép kết nối an toàn (SSL)

},

// =======================

// KHÓA MÃ HÓA

// =======================

"Encryption": {

"Key": "Your-Encryption-Key-Here"

// 🔹 Key này dùng để mã hóa dữ liệu nhạy cảm (ví dụ: mật khẩu, số thẻ)

},

// =======================

// BÁO CÁO TÀI CHÍNH

// =======================

"FinancialReporting": {

"IntervalMinutes": 1440, // Bao lâu chạy job (1440 phút = 1 ngày)

"ReportType": "Monthly", // Loại báo cáo: Daily / Weekly / Monthly / Custom

"CustomDays": 30, // Nếu chọn "Custom" thì sẽ dùng số ngày này

"Enabled": true // Bật/tắt tính năng báo cáo

},

// =======================

// EMAIL THÔNG BÁO

// =======================

"Email": {

"UseAdvancedNotifications": true, // Có dùng cơ chế notify nâng cao không

"DryRun": false, // Nếu true: chỉ test, không gửi thật

"SmtpHost": "smtp.yourmail.com", // SMTP server

"SmtpPort": 587, // Cổng gửi mail (587 = TLS, 465 = SSL)

"FromEmail": "noreply@yourcompany.com", // Email gửi đi

"FromName": "Your System Name", // Tên hiển thị

"Username": "your-username", // User SMTP

"Password": "your-password", // Password hoặc app password

"AppPassword": "your-app-password", // Gmail/Yahoo/... dùng app password

"AlertRecipients": [

"admin@yourcompany.com" // Danh sách email nhận cảnh báo

]

},

// =======================

// EMAIL ĐƠN GIẢN (Test / Dev)

// =======================

"EmailSimple": {

"UsePickupDirectory": true, // Ghi mail ra file thay vì gửi thật

"PickupDirectory": "C:\\temp\\emails", // Thư mục lưu mail

"AlertsFromAddress": "noreply@yourcompany.com",

"AlertsToAddress": "admin@yourcompany.com",

"Smtp": {

"Host": "smtp.yourmail.com",

"Port": 587,

"EnableSsl": true,

"Username": "your-username",

"Password": "your-password"

}

},

// =======================

// WORKER KIỂM TRA TỒN KHO

// =======================

"StockWorker": {

"CheckIntervalMinutes": 5, // Bao lâu thì check tồn kho (5 phút/lần)

"LowStockThreshold": 10, // Ngưỡng cảnh báo hết hàng sớm

"CriticalStockThreshold": 5 // Ngưỡng cảnh báo khẩn cấp

},

// =======================

// LOGGING CƠ BẢN

// =======================

"Logging": {

"LogLevel": {

"Default": "Information", // Mức log mặc định

"System": "Warning", // Hệ thống: chỉ log Warning+

"Microsoft": "Warning", // Microsoft lib: Warning+

"Microsoft.Hosting.Lifetime": "Information",

"Dashboard.StockWorker": "Debug" // Riêng StockWorker log chi tiết

}

},

// =======================

// SERILOG (LOG NÂNG CAO)

// =======================

"Serilog": {

"Using": [ "Serilog.Sinks.Console", "Serilog.Sinks.File" ],

"MinimumLevel": {

"Default": "Debug", // Log mặc định chi tiết

"Override": {

"Microsoft": "Warning", // Giảm log của Microsoft

"System": "Warning"

}

},

"WriteTo": [

{

"Name": "Console" // Ghi log ra console

},

{

"Name": "File", // Ghi log ra file

"Args": {

"path": "logs/log-.txt", // Đường dẫn log

"rollingInterval": "Day", // Tạo file log mới mỗi ngày

"rollOnFileSizeLimit": true, // Tự tạo file mới khi quá dung lượng

"shared": true,

"retainedFileCountLimit": 7, // Giữ lại tối đa 7 file log

"fileSizeLimitBytes": 10485760 // Giới hạn file ~10MB

}

}

]

},

// =======================

// BẢO MẬT & JWT

// =======================

"Security": {

"PasswordSalt": "Your-Salt-Here", // Salt để hash password

"EncryptionKey": "base64-key-32bytes==", // Key mã hóa

"EncryptionIV": "base64-iv-16bytes==", // IV cho AES/GCM

"Jwt": {

"Issuer": "YourAppName", // Ai phát hành token

"Audience": "YourAudience", // Ai được chấp nhận token

"SecretKey": "your-long-secret-key", // Key bí mật để ký JWT

"ExpirationMinutes": 60 // Token hết hạn sau bao lâu

}

},

// =======================

// CẤU HÌNH HOST

// =======================

"AllowedHosts": "\*" // Cho phép tất cả host truy cập (dev). Prod nên giới hạn

}

Sau đó vào visual studio

A screenshot of a computer

AI-generated content may be incorrect.

Trên thanh head bar có icon bánh răng, click vào đó chọn Stock worker và thực hiện run dự án như bình thường.

**Đối với dự án Dashboard**

Tiếp tục vào Dashboard.Winform/appsetting.json

Thực hiện bước config connection string và secret key cho database dựa theo hướng dẫn bên dưới.

{

// =======================

// SERILOG - QUẢN LÝ LOG

// =======================

"Serilog": {

"Using": [ "Serilog.Sinks.Console", "Serilog.Sinks.File" ],

// 🔹 Danh sách các "sink" (nơi ghi log). Ở đây gồm console & file.

"MinimumLevel": {

"Default": "Debug", // Ghi log chi tiết mặc định (Debug, Info, Warn, Error)

"Override": {

"Microsoft": "Warning", // Giảm log của thư viện Microsoft

"System": "Warning" // Giảm log của thư viện System

}

},

"WriteTo": [

{

"Name": "Console"

// 🔹 Ghi log trực tiếp ra console

},

{

"Name": "File",

"Args": {

"path": "logs/log-.txt", // Đường dẫn file log (có rolling file theo ngày)

"rollingInterval": "Day", // Tạo log file mới mỗi ngày

"rollOnFileSizeLimit": true, // Nếu file quá lớn thì tự cắt file mới

"shared": true, // Cho phép nhiều process ghi log vào cùng file

"retainedFileCountLimit": 7, // Chỉ giữ tối đa 7 file log (dọn dẹp file cũ)

"fileSizeLimitBytes": 10485760 // Giới hạn file log = 10 MB

}

}

]

},

// =======================

// BẢO MẬT & JWT

// =======================

"Security": {

"PasswordSalt": "CoffeeShop2024",

// 🔹 Salt được thêm vào khi hash password để tăng tính bảo mật

"EncryptionKey": "base64-key-32bytes==",

// 🔹 Key 32 bytes (Base64) dùng cho AES hoặc thuật toán mã hóa đối xứng

"EncryptionIV": "base64-iv-16bytes==",

// 🔹 IV (Initialization Vector) 16 bytes, dùng trong AES CBC/GCM

"Jwt": {

"Issuer": "DashboardApp", // Ai phát hành token (application)

"Audience": "DashboardUsers", // Ai có quyền sử dụng token này

"SecretKey": "this\_is\_a\_very\_long\_secret\_key\_123456", // Key bí mật để ký JWT

"ExpirationMinutes": 60 // Token sẽ hết hạn sau 60 phút

}

},

// =======================

// CẤU HÌNH HOST

// =======================

"AllowedHosts": "\*",

// 🔹 "\*" = cho phép mọi host truy cập (dùng dev/test).

// 🔹 Prod nên giới hạn cụ thể host/domain.

// =======================

// KẾT NỐI DATABASE

// =======================

"ConnectionStrings": {

"WebbanhangDB": "Server=127.0.0.1,1433;Database=webbanhang;User Id=root;Password=db\_password@123;encrypt=true;trustServerCertificate=true"

// 🔹 Kết nối tới SQL Server (localhost:1433)

// 🔹 Database = webbanhang

// 🔹 User Id = root, Password = db\_password@123

// 🔹 encrypt=true + trustServerCertificate=true: hỗ trợ SSL nhưng chấp nhận cert tự ký

},

// =======================

// KHÓA MÃ HÓA (ỨNG DỤNG)

// =======================

"Encryption": {

"Key": "DayLachuOiKeySieUDaiMaDuyDEpTRaiBoVao@12331238182373787234"

// 🔹 Key riêng biệt để app mã hóa/giải mã dữ liệu (ví dụ: thông tin nhạy cảm trong DB)

}

}

Lưu ý cấu hình File sql bao gồm 3 file

init\_db.sql // tạo database và login  
init\_schema.sql // thực hiện tạo schema database   
init\_seed\_data.sql // thực hiện seed dữ liệu ảo